

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Conception et réalisation d'une borne interactive multimédia

Deflorenne, Arnaud; Mathot, Patrice

Award date:
1995

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre Dame de la Paix
Institut d'Informatique
Rue Grandgagnage, 21
B 5000 - Namur (Belgium)

Conception et réalisation
d'une borne interactive multimédia

Mémoire présenté en vue de l'obtention du
grade de Licencié et Maître en Informatique

par
Arnaud Deflorenne et Patrice Mathot

Promoteur: Monsieur Roland Lesuisse

Année académique 1994-1995

Résumé

Ce mémoire présente le développement complet d'une borne interactive multimédia servant à présenter des universités.

Il présente une série de concepts théoriques illustrés à chaque fois par le logiciel que nous avons réalisé.

Nous expliquons d'abord ce que doit contenir un cahier des charges standard, établissons la liste des méthodologies qui peuvent être envisagées pour la conception du logiciel et expliquons celle que nous avons retenue.

Ensuite, nous définissons le concept de scénario en informatique, en ayant recours au secteur cinématographique et publicitaire.

Après cela, nous présentons les règles ergonomiques à respecter lors de l'élaboration d'interfaces et terminons par la réalisation de la borne, c'est-à-dire les différentes étapes par lesquelles nous sommes passés pour aboutir au produit final.

Abstract

This thesis presents the complete development of a multimedia kiosk used for giving a description of universities.

It sets out a series of theoretical concepts, each of which being illustrated by the software we have made.

We first explain what a standard requirements document should contain, we draw up a list of the methodologies that can be envisaged for the conception of the software and we explain the one we have chosen.

We then define the concept of scenario in computer science resorting to the fields of cinema and advertising.

Afterwards, we present the ergonomic rules that must be respected when working out interfaces and in the end we explain the making of the kiosk i.e. the different stages we have gone through to achieve the final product.

Remerciements

Nous tenons à remercier plusieurs personnes pour le soutien qu'elles nous ont apporté lors de l'élaboration de ce travail. Pour commencer, nous remercions Monsieur LESUISSE qui nous a guidés dans ce mémoire. Nous remercions également les différents professeurs et assistants de l'institut d'informatique.

En ce qui concerne notre stage à Strasbourg, nous sommes tout particulièrement reconnaissants à Madame LEDIG ainsi qu'à toute son équipe pour leurs conseils et pour leur accueil à l'Association pour le Développement des Entreprises et des Compétences du Val de Moder. Nous voulons enfin remercier Monsieur NANOPOULOS pour son aide tout au long du développement de notre projet.

Table des matières

1. INTRODUCTION	1
2. LES BORNES INTERACTIVES MULTIMÉDIA	3
2.1. LES BORNES INTERACTIVES	3
2.1.1. UN PEU D'HISTOIRE: AVANT LES BORNES INTERACTIVES...	3
2.1.2. LES BORNES INTERACTIVES: UN NOUVEAU MOYEN DE COMMUNICATION?	4
a. La communication	4
b. Les bornes interactives	4
2.1.3. LES PARTICULARITÉS DES BORNES INTERACTIVES	5
2.2. LE MULTIMÉDIA ET LES LOGICIELS MULTIMÉDIA	6
2.2.1. LE MULTIMÉDIA	6
a. Définition	6
b. Les éléments du multimédia et leurs relations	7
2.2.2. LES LOGICIELS MULTIMÉDIA	8
a. Définition	8
b. Caractéristiques	9
c. Le marché des logiciels multimédia	10
3. LE CAHIER DES CHARGES DE TÉLÉMAQUE	11
3.1. LA CONCEPTION STANDARD D'UN CAHIER DES CHARGES	11
3.1.1. SPÉCIFICATIONS DES BESOINS DU CLIENT	11
3.1.2. PLAN DE DÉVELOPPEMENT	12
3.1.3. PLAN QUALITÉ	13
3.2. APPLICATION À NOTRE BORNE INTERACTIVE	13
3.2.1. LES CONTRAINTES FONCTIONNELLES	13
3.2.2. LES CONTRAINTES NON-FONCTIONNELLES	14
a. Taux de panne	14
b. Possibilité de récupération	14
c. Facilité d'utilisation	14
d. Performances	15
e. Maintenance	15
f. Portabilité	15
g. Sécurité	15
4. LA MÉTHODOLOGIE	17
4.1. ETUDE DES MÉTHODOLOGIES EXISTANTES	17
4.1.1. MODÈLE DE LA CASCADE	17
4.1.2. L'APPROCHE PAR PROTOTYPAGE	18

4.1.3. L'APPROCHE OPÉRATIONNELLE	19
4.1.4. L'APPROCHE MIXTE	20
4.2. LA MÉTHODOLOGIE UTILISÉE POUR TÉLÉMAQUE	22
4.2.1. LE PROTOTYPE NON-OPÉRATIONNEL	23
4.2.2. SIMULATIONS RAPIDES: DÉVELOPPEMENT DES INTERFACES	24
4.2.3. DÉMONSTRATIONS RAPIDES: DÉVELOPPEMENT DE LA BASE DE DONNÉES	24
4.2.4. TESTS SUPPLÉMENTAIRES	24
4.3. JUSTIFICATIONS DE LA MÉTHODOLOGIE CHOISIE	24
 5. LE SCÉNARIO	 27
5.1. RAPPORTS ENTRE SCÉNARIO CINÉMATOGRAPHIQUE ET SCÉNARIO INFORMATIQUE	27
5.1.1. PREMIÈRE DÉFINITION D'UN SCÉNARIO	27
5.1.2. LES ÉLÉMENTS DU SCÉNARIO	28
a. Histoire et narration	28
b. L'unité de sujet (intrigue principale)	28
c. L'intrigue secondaire	29
d. Le dialogue	30
5.1.3. LES PARTIES DU SCÉNARIO	31
a. L'exposition	31
b. Le climax	31
c. La loi de progression continue	32
d. Scènes et séquences	32
e. L'accroche	33
5.1.4. LES PROCÉDÉS DE NARRATION	33
a. Règles de la dramatisation	33
b. Informations	34
5.2. RAPPORTS ENTRE SCÉNARIO PUBLICITAIRE ET SCÉNARIO INFORMATIQUE	35
5.2.1. RÔLE DU SCÉNARIO	35
5.2.2. DÉFINITION DES BESOINS	36
5.2.3. ANALYSE DU PRODUIT ET DU CLIENT	36
5.2.4. VALIDATION PAR LE CLIENT	37
5.3. LE SCÉNARIO INFORMATIQUE	37
5.3.1. DÉFINITIONS	37
5.3.2. CARACTÉRISTIQUES	38
5.3.3. LE SCÉNARIO DE TÉLÉMAQUE	39
a. Apports du monde publicitaire	39
b. L'analyse de la tâche	39
c. Apports du monde cinématographique	41
d. Caractéristiques du scénario de notre borne	42
 6. LES INTERFACES DE TÉLÉMAQUE	 45
6.1. RAPPEL DE QUELQUES NOTIONS D'ERGONOMIE	45
6.1.1. QU'EST-CE QU'UNE INTERFACE?	45
6.1.2. MODÈLE DE NORMAN	46
6.1.3. DISTANCES CONTENUES DANS LE LANGAGE DE L'INTERFACE	49

a. Distance sémantique	49
b. Distance articulatoire	50
6.1.4. DEGRÉ D'IMPLICATION DIRECTE DE L'UTILISATEUR	50
a. Métaphore du mini-monde	51
b. Métaphore de la conversation	51
c. La métaphore navigationnelle	51
6.2. ETUDE DES ÉCRANS DE TÉLÉMAQUE	51
6.2.1. LES MOYENS D'INTERACTION	52
6.2.2. DÉFINITION DES PRINCIPAUX OBJETS INTERACTIFS ABSTRAITS DE TÉLÉMAQUE	53
a. Icône	53
b. Menu	53
c. Bouton de commande	53
d. Bouton graphique	53
e. Liste de sélection	53
6.2.3. CRITÈRES ERGONOMIQUES DE DESIGN	53
a. Définitions et pertinence	54
b. Télémaque et les critères de design	57
c. Quelques autres repères ergonomiques	58
6.2.4. PRÉSENTATION DES DIFFÉRENTS TYPES D'ÉCRANS	59
a. Les écrans de choix	59
b. Les écrans avec informations terminales	60
c. Les écrans de transition	60
6.2.5. LES CARACTÉRISTIQUES COMMUNES DES ÉCRANS DE TÉLÉMAQUE	60
a. L'utilisation des couleurs	60
b. La découpe des écrans	61
6.2.6. ANALYSE DES CARACTÉRISTIQUES SPÉCIFIQUES PAR TYPE D'ÉCRAN	65
a. Les écrans de choix	65
b. Les écrans avec informations terminales	71
c. Les écrans de transition	75
6.3. EVALUATION DES ÉCRANS DE TÉLÉMAQUE	76
6.3.1. LE TEMPS D'APPRENTISSAGE	76
6.3.2. LA RAPIDITÉ D'EXÉCUTION D'UNE TÂCHE	76
6.3.3. LE TAUX D'ERREURS EFFECTUÉES PAR L'UTILISATEUR	76
6.3.4. LA PÉRIODE DE RÉMANENCE	76
6.3.5. LA SATISFACTION SUBJECTIVE À UTILISER LE SYSTÈME	77
6.3.6. LA COUVERTURE DES COMMANDES	77
6.4. GESTION DES DIFFÉRENTS MÉDIA	77
6.4.1. LE TEXTE	77
6.4.2. LE SON	77
a. La voix	77
b. La musique	78
6.4.3. L'IMAGE FIXE	78
a. Les graphes	78
b. Les photos	78
c. Le logo	79
6.4.4. L'IMAGE ANIMÉE	79
6.4.5. INTÉGRATION DES MÉDIA	79
a. Modèles conceptuels du temps	80
b. Modèles conceptuels de la synchronisation	81
c. Modèle de visualisation	83

7. RÉALISATION DE LA BORNE	85
7.1. LA PLATE-FORME DE DÉVELOPPEMENT	85
7.1.1. CHOIX DE L'ENVIRONNEMENT MATÉRIEL	85
7.1.2. CHOIX DE L'ENVIRONNEMENT LOGICIEL	86
a. Choix du langage de développement	86
b. Choix du système de gestion de base de données	89
c. Choix du langage de prototypage	89
7.2. LES DIFFÉRENTES ÉTAPES DE LA CONCEPTION	90
7.2.1. EVALUATION DES RISQUES	90
7.2.2. RÉALISATION DU PROTOTYPE NON-OPÉRATIONNEL	91
7.2.3. SIMULATIONS RAPIDES: DÉVELOPPEMENT DES INTERFACES	91
a. Implantation des interfaces	91
b. Difficultés rencontrées	91
7.2.4. DÉMONSTRATIONS RAPIDES: DÉVELOPPEMENT DE LA BASE DE DONNÉES	94
a. Réalisation des différents schémas	94
b. Implantation de la base de données	100
c. Modification du code	100
7.2.5. TESTS SUPPLÉMENTAIRES	100
a. Les testeurs	101
b. La procédure de test	101
8. CONCLUSION	105
9. BIBLIOGRAPHIE	107
10. ANNEXES	111

Table des figures

FIGURE 2-1. SCHÉMA NORMAL DE LA COMMUNICATION	4
FIGURE 4-1. MODÈLE DE LA CASCADE	18
FIGURE 4-2. L'APPROCHE PAR PROTOTYPAGE	19
FIGURE 4-3. L'APPROCHE OPÉRATIONNELLE	20
FIGURE 4-4. L'APPROCHE MIXTE	21
FIGURE 4-5. DESIGN ITÉRATIF	22
FIGURE 4-6. LA MÉTHODOLOGIE RETENUE	23
FIGURE 5-1. GRAPHE DE TRANSITION DE TÉLÉMAQUE	43
FIGURE 6-1. MODÈLE DE NORMAN APPLIQUÉ À UNE TÂCHE DE TÉLÉMAQUE	48
FIGURE 6-2. L'ÉCRAN "CHOIX DU PAYS"	48
FIGURE 6-3. DISTANCES SÉMANTIQUE ET ARTICULATOIRE	49

FIGURE 6-4. L'ÉCRAN "PRÉSENTATION DE LA VILLE"	61
FIGURE 6-5. DÉCOUPE DES ÉCRANS DE TÉLÉMAQUE	62
FIGURE 6-6. LA ZONE DE TITRE DE L'ÉCRAN "CHOIX DE LA MATIÈRE"	62
FIGURE 6-7. LE BOUTON GRAPHIQUE "RETOUR AU MENU PRINCIPAL"	63
FIGURE 6-8. LE BOUTON GRAPHIQUE "RETOUR À L'ÉCRAN PRÉCÉDENT"	63
FIGURE 6-9. LE BOUTON GRAPHIQUE "HISTORIQUE"	63
FIGURE 6-10. LE BOUTON GRAPHIQUE "CHANGEMENT DE LANGUE"	64
FIGURE 6-11. LE BOUTON GRAPHIQUE "CHANGEMENT D'UNIVERSITÉ"	64
FIGURE 6-12. LE BOUTON GRAPHIQUE "AIDE"	64
FIGURE 6-13. LE BOUTON GRAPHIQUE "FIN"	65
FIGURE 6-14. L'ÉCRAN DE PRÉSENTATION	66
FIGURE 6-15. L'ÉCRAN "MENU PRINCIPAL"	67
FIGURE 6-16. L'ÉCRAN "CHOIX DU PAYS"	68
FIGURE 6-17. L'ÉCRAN "CHOIX DE LA MATIÈRE"	69
FIGURE 6-18. POSITIONNEMENT DES BOUTONS DE COMMANDE	69
FIGURE 6-19. L'ÉCRAN "CHANGEMENT D'UNIVERSITÉ"	70
FIGURE 6-20. L'ÉCRAN "HISTORIQUE"	71
FIGURE 6-21. L'ÉCRAN "PRÉSENTATION DE LA VILLE"	72
FIGURE 6-22. L'ÉCRAN "PRÉSENTATION DE STRASBOURG"	72
FIGURE 6-23. L'ÉCRAN "COURS"	73
FIGURE 6-24. L'ÉCRAN "MUSÉES"	73
FIGURE 6-25. LES FLÈCHES	74
FIGURE 6-26. PLAN DE LA VILLE	74
FIGURE 6-27. IMPRESSION	74
FIGURE 6-28. L'ÉCRAN "QUELQUES CHIFFRES"	75
FIGURE 6-29. L'ÉCRAN "INTRODUCTION DE LA VILLE"	75
FIGURE 6-30. LOGO DE TÉLÉMAQUE	79
FIGURE 6-31. RELATIONS TEMPORELLES BINAIRES	80
FIGURE 6-32. REPRÉSENTATION DE LA PRÉSENTATION DES VILLES ET DES UNIVERSITÉS SELON LES INTERVALLES TEMPORELS	81
FIGURE 6-33. TRANSFORMATION NON TEMPORELLE AU NIVEAU DE LA VIDÉO	83
FIGURE 6-34. UTILISATION D'UN RÉSEAU DE PÉTRI POUR REPRÉSENTER LA PRÉSENTATION DES VILLES ET DES UNIVERSITÉS	84
FIGURE 7-1. LE SCHÉMA CONCEPTUEL	96
FIGURE 7-2. SCHÉMA CONFORME AU MODÈLE RELATIONNEL	99

Table des tableaux

TABEAU 6-1. L'ASSOCIATION DES COULEURS	60
TABEAU 7-1. ETUDE DU LANGAGE DE DÉVELOPPEMENT	89
TABEAU 7-2. IDENTIFIANTS DU SCHÉMA CONCEPTUEL	95

1. Introduction

Le produit qu'il nous a été demandé de réaliser, dans le cadre de ce travail, est une borne interactive multimédia. Celle-ci, placée à l'IECS¹, doit être capable de donner aux étudiants de cette institution tous les renseignements dont ils ont besoin sur la ville et sur l'université de Strasbourg ainsi que sur sa cinquantaine d'universités partenaires réparties à travers le monde. Cette borne, appelée Télémaque, doit pouvoir être consultée et mise à jour par des utilisateurs n'ayant aucune connaissance en informatique.

Ce mémoire traite donc du développement complet d'un logiciel, depuis le cahier des charges jusqu'à l'implémentation.

Les termes "borne interactive" et "multimédia" sont expliqués dans le chapitre 2. Nous montrerons dans ce chapitre, et tout au long du mémoire, que ces deux termes, s'ils sont souvent utilisés ensemble, ont des effets différents sur la conception d'un logiciel.

Les spécifications du problème qu'il nous a été demandé de résoudre sont explicitées en détail au chapitre 3. Après plusieurs rencontres avec le client, nous avons élaboré une première version du cahier des charges du logiciel, ce qui nous a permis d'identifier un certain nombre de contraintes.

Une fois le cahier des charges rédigé, nous avons essayé de trouver une solution au problème posé. Pour ce faire, nous avons élaboré une méthodologie qui sera suivie pendant toute la conception. La méthodologie que nous avons décidé d'adopter pour notre logiciel est présentée au chapitre 4.

La définition de la méthodologie a permis de mettre en évidence deux points cruciaux dans le développement de toute borne interactive multimédia.

Le premier de ces points, expliqué au chapitre 5, est la conception d'un scénario. Celui-ci concerne la méthode adoptée pour restituer l'information à l'utilisateur. Nous montrerons que le rôle joué par le scénario en informatique n'est pas éloigné de celui qu'il joue dans le secteur qui l'a vu naître: le cinéma.

Le chapitre 6, quant à lui, présentera le deuxième point crucial dans la construction d'une borne interactive: les interfaces. Nous présenterons dans ce chapitre les règles ergonomiques classiques à respecter ainsi que celles propres aux bornes interactives multimédia.

¹ *Institut des études de commerce de Strasbourg.*

Le dernier chapitre (le chapitre 7) est consacré à la réalisation de la borne proprement-dite. Nous y mettons en exergue les différentes étapes qui ont été nécessaires pour aboutir au produit final en partant du cahier des charges et en suivant la méthodologie retenue.

Dans la conclusion, nous précisons quel était notre objectif, ce que nous avons fait pour l'atteindre et quels sont les développements futurs de Télémaque.

Le code du programme, le scénario et le guide d'exploitation de la base de données peuvent être consultés dans les annexes.

2. Les bornes interactives multimédia

Dans ce chapitre, nous allons introduire les deux concepts les plus importants de ce mémoire: les bornes interactives (section 2.1) et le multimédia (section 2.2). Si nous les présentons séparément, c'est pour souligner que, même si ces concepts sont souvent utilisés conjointement, ils désignent des réalités bien différentes.

2.1. Les bornes interactives

Avant de présenter une définition d'une borne interactive, nous allons retracer son évolution. Nous terminerons cette section en citant ses principales caractéristiques.

2.1.1. Un peu d'histoire: avant les bornes interactives...

Les bornes interactives sont très à la mode pour l'instant. Tout endroit (privé ou public), en contact avec des personnes, veut sa borne: musées, gares des chemins de fer, offices du tourisme, grosses entreprises... Avec le temps, ces bornes ont évolué.

D'après [PIC93], on peut dégager trois temps dans la genèse des bornes interactives telles que nous les connaissons actuellement. Tout d'abord au début des années 80, sont apparues des bornes interactives qui permettaient l'échange d'informations entre un "spectateur" et des vidéogrammes. Jusqu'au début des années 1990, l'interaction était assez limitée et l'information était surtout transmise aux utilisateurs par des séquences audiovisuelles assez longues. Dans un deuxième temps (tout au long des années 80), les technologies numériques se sont fortement développées. De ce fait, les micro-ordinateurs sont entrés dans les studios de montage et petit à petit, l'interaction a pris la place que nous lui connaissons actuellement¹. Le troisième volet de ce développement est, actuellement, l'arrivée de nouveaux acteurs dans le monde de la conception de bornes interactives. Ceux-ci sont principalement des informaticiens maîtrisant les techniques multimédia ou des artistes connaissant la technique informatique.

Après avoir brossé un bref aperçu historique de l'évolution des bornes interactives, tentons d'en donner une définition.

¹ Une seule ombre au tableau pour les éditeurs: l'hétérogénéité des matériels utilisés interdit une exploitation commerciale importante.

2.1.2. Les bornes interactives: un nouveau moyen de communication?

Cette section, qui a été écrite en se basant sur [PIC93], présente d'abord le schéma normal de la communication avant de l'appliquer aux bornes interactives et d'en donner de la sorte une définition.

a. La communication

Qui dit bornes interactives, dit communication. S'il s'agit d'une communication un peu particulière (entre un homme et une machine), elle peut quand même être modélisée selon le même schéma que la communication humaine. Ce schéma apparaît à la Figure 2-1.

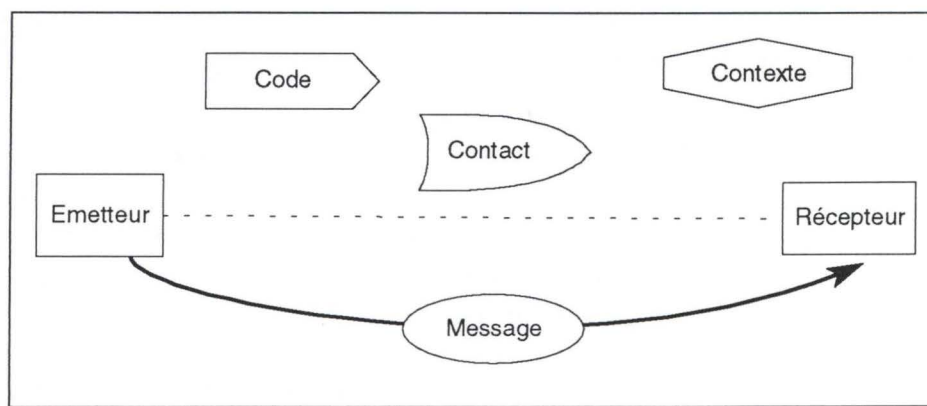


Figure 2-1. Schéma normal de la communication

Comme on peut le voir, un émetteur envoie un message à un récepteur. Pour se comprendre, ces deux entités doivent utiliser le même code, le même langage. Pour donner un sens au message, il faut un contexte qui permettra donc d'interpréter le message correctement. Enfin pour que cette communication puisse s'établir, il faut bien entendu un canal de communication (la parole, l'écriture, etc.): c'est ce qu'on appelle le contact.

b. Les bornes interactives

A partir de ce schéma, on peut définir une borne interactive comme un automate interactif qui offre de l'information à tout être humain qui s'en approche. En d'autres termes, une borne interactive est une nouvelle technologie de distribution de l'information. Cette définition convient tout à fait à la borne qui a été créée pour les jeux olympiques de Los Angeles en 1984. Celle-ci, appelée "The Olympic Message System", permettait aux athlètes de rester en contact avec leur famille. Avant, ce contact était maintenu grâce au téléphone. Cela impliquait que les familles pouvaient téléphoner pour rien (l'athlète n'était pas joignable). A Los Angeles, par contre, tout coup de téléphone était reçu car enregistré dans la borne. Les athlètes

pouvaient aller les écouter quand ils le voulaient. Cette borne est donc bien une nouvelle technologie de distribution de l'information.

Comme le laisse présager le schéma de la communication, on peut constater que deux entités sont constamment présentes: un être humain (l'utilisateur) et une machine (la borne interactive). Ce nouveau mode de communication présente quelques caractéristiques intéressantes.

2.1.3. Les particularités des bornes interactives

Les bornes interactives ont des caractéristiques distinctives dont deux nous paraissent importantes.

1. Les personnes, qui utilisent les bornes interactives, sont en général inexpérimentées face à de telles machines. Elles n'ont aucune compétence technique et en plus, elles utilisent la borne de manière très occasionnelle.
2. La motivation de ces utilisateurs est diverse. En effet, face à certaines bornes telles que des bornes de marketing, les utilisateurs n'auront pas la même motivation que devant un automate leur permettant de trouver des informations capitales pour eux ou d'acquérir un billet de train. Ce fait est bien mis en évidence par [OLI93]. « *The multimedia kiosks are stations that offer services as a result of a dialog session with a user. Take for example, a kiosk that presents a catalogue of products that may be purchased by means of credit card. This kiosk has two uses as regards the user:*

1) It permits him to purchase,

2) It can sell the products to him.

'Permit to purchase' is the final part of the dialogue: user has already decided to buy a certain product.(...)

However, 'selling', is another operation. The user in this case does not have a goal; he is not motivated. The only thing that creates the dialog session and keeps it going is his curiosity.(...) The problem is, therefore, to keep the user's attention fixed for the necessary time by presenting the alternatives in a clear and pleasant manner by making sure that the potential client still wants to 'see something new' ».

Ces caractéristiques auront leur importance tant dans la conception du scénario que dans celle des interfaces, comme nous le verrons ultérieurement.

Bien que ce ne soit pas une implication nécessaire, qui dit borne interactive dit utilisation du multimédia. Il convient donc que nous précisions ce concept un peu vague.

2.2. Le multimédia et les logiciels multimédia

2.2.1. Le multimédia

a. Définition

« *"Vous êtes sur la plage..., il fait beau..., le ciel est bleu..., le soleil radieux..., la température idéale, vous rêvez en fixant l'horizon... . Tout à coup au loin une tache sombre apparaît, elle grandit à vue d'oeil, vous sortez de votre torpeur, elle arrive sur vous à une vitesse extraordinaire, c'est une tempête, qui se transforme en un ouragan d'une force sans précédent. Vous devez faire face, vous devez vous adapter à cette révolution des forces de la nature. Mesdames et Messieurs, la tache sombre qui arrive au loin c'est le multimédia et nous sommes ici pour nous préparer à la mutation qu'il va engendrer"*. C'est par ces mots que le président d'une des plus grandes compagnies d'informatique du monde a récemment commencé une journée d'étude stratégique du groupe. » [GOR95]

Si on cherche une définition formelle de ce terme, on peut diviser "multimédia" en deux parties. "Multi" signifie plusieurs et "média" représente « *tout support de diffusion de l'information constituant à la fois un moyen d'expression et un intermédiaire transmettant un message à l'intention d'un groupe* ». [CRA94]

Il n'existe pas encore un réel consensus autour de la définition du multimédia. Pour Goray, le multimédia est « *l'association de plusieurs monomédia avec description des liens logiques qui lient les éléments d'informations monomédia* » [GOR95]. Si cette définition ne tient compte que des média en sortie, le multimédia pour Vaananen concerne aussi bien les outputs que les inputs: « *du point de vue de l'utilisateur, le multimédia c'est utiliser plusieurs média en entrée (texte, parole, manipulation directe) et plusieurs média en sortie (texte, son, graphique, vidéo)* ». [VAA93]

Pour notre part, nous nous en tiendrons dans la suite du travail à la définition de Goray [GOR95].

S'il est vrai qu'on entend le terme multimédia un peu partout, il est également vrai que ce terme est parfois utilisé à tort et à travers. On le confond notamment avec un autre terme à la mode: les autoroutes de l'information. Si elles voient le jour, les autoroutes de l'information seront le canal de communication du multimédia. On pourrait dire qu'il s'agit d'un réseau numérique véhiculant différents types d'information dans les deux sens et à haut débit. Cela signifie, par exemple, que les chaînes de télévision, le téléphone et les données purement informatiques seraient véhiculées sur un même réseau et à la vitesse de plusieurs centaines de

Mégabits par seconde. On voit que le contenu des autoroutes sera aussi bien du texte et des images que du son et de la vidéo donc on peut parler de canal de communication du multimédia.

b. Les éléments du multimédia et leurs relations

Selon notre définition, le multimédia est « *l'association de plusieurs monomédia avec description de liens logiques qui lient ces éléments d'information* ». Nous allons, dans ce paragraphe, passer en revue les différents types de données que peut contenir une information multimédia [GRE93] ainsi que leurs relations (couplage) avec le temps, l'espace, l'utilisateur et les autres média [CRA94].

(1) Le texte

C'est le médium traditionnel de l'informatique. Il est caractérisé par un couplage spatial: il est en relation avec l'espace dans lequel il s'inscrit. Ce couplage est monodirectionnel (le texte a besoin de l'espace pour exister mais le contraire n'est pas vrai) et orienté (il y a un ordre de lecture d'un texte pour que celui-ci soit compréhensible). L'espace peut également être multicouplé c'est à dire que plusieurs textes se partagent l'espace disponible même si, à un moment donné, l'utilisateur ne sera concentré que sur un seul de ces textes. Cette relation entraîne certaines conséquences auxquelles il faudra faire attention: la taille des caractères, la densité du texte, le partage clair de l'espace (si plusieurs textes cohabitent)... La relation du texte avec le temps n'existe pas sauf dans le cas où le texte est défilant. Cette exception est également valable pour le couplage avec l'utilisateur où le texte est normalement passif et non synchronisé avec l'activité de lecture.

(2) Le son

Le son peut être de différentes natures: les bruits, la voix, la musique. Le couplage au temps est évident: le son se déroule et évolue (ou n'évolue pas) au fil du temps. La relation avec l'espace est, elle, moins importante (le son n'a pas besoin de l'espace pour exister). Le cas de l'utilisateur est, quant à lui, très intéressant. L'éloignement joue un rôle capital: si l'utilisateur est trop loin, il n'entendra pas le son émis. Par contre, la position de la personne importe peu: elle ne doit pas, comme c'est le cas pour le texte, être face à la source. L'espace peut être multicouplé (plusieurs sons simultanés) mais il faut tenir compte de la capacité de discernement de l'utilisateur qui n'est pas infinie. En paraphrasant une expression célèbre, nous pourrions dire: « trop de sons tuent le son ». La combinaison texte-son est pertinente si le son est une musique mais elle est à éviter si le son est porteur de sens: à un moment donné, l'utilisateur devra faire un choix entre lire et écouter.

(3) Les graphiques

L'image fixe

Les éléments représentatifs de cette catégorie sont les graphes, les images scannées (les photos), les images capturées (des images prises d'une séquence vidéo ou d'un film) .

Le couplage avec l'espace est proche de celui décrit pour le texte, la différence résidant dans le type d'information présenté. Si le mode de lecture est différent (une personne ne lit pas une photo comme elle lit un texte), la relation avec l'utilisateur est identique. Le couplage au temps peut exister si l'image est décomposable en éléments décrits les uns après les autres par un commentaire. L'introduction du médium son est alors nécessaire et même souhaitée dans le cas de commentaire ou de musique. Dans les autres cas, comme pour le texte, il faut éviter l'utilisation de deux média conflictuels. L'utilisation combinée du texte et de l'image peut être bénéfique si elle est faite correctement: ces deux média ont un couplage avec l'espace et leur combinaison doit respecter les capacités de perception de l'utilisateur (un seul médium peut être perçu à la fois).

L'image animée

Dans cette catégorie, figurent la vidéo et les animations.

La différence entre l'image fixe et l'image animée est la relation qu'elles ont avec le temps. L'image animée ne prend tout son sens que dans sa mobilité, dans son rapport avec les images précédentes et suivantes qui créent un contexte à un moment donné. Le son entretient le même rapport avec le temps: le couplage image animée-son peut donc être réalisé si le choix des instances de ces deux média est pertinent. La différence qui existe est que, lorsqu'il est arrêté, le son perd tout son sens alors que l'image animée, elle, n'est pas dénuée de tout sens lorsqu'elle est suspendue: elle se lit autrement car elle est retirée de son contexte.

On peut se demander comment l'ordinateur opère pour traiter des types d'informations aussi différents que ceux qui viennent d'être présentés. Il le fait grâce à la numérisation de l'information qui, mise sous une forme commune, peut alors être traitée.

Après l'énumération des différents types de données que couvre le multimédia, il nous semble intéressant de préciser ce que n'est pas le multimédia.

2.2.2. Les logiciels multimédia

a. Définition

Pour Vanderdonckt, « *un logiciel est multimédia lorsqu'il utilise plus de deux média et qu'il n'y a pas prédominance du texte* » [VAN95]. Cette définition est assez restrictive puisque le multimédia ne commence qu'à partir de trois média et que, de plus, le texte ne doit pas être

prédominant. Cette définition a pour but d'exclure des logiciels multimédia les programmes présentant du texte accompagné d'un bip occasionnel.

Blattner et Dannenberg disent que « *un logiciel multimédia possède plus d'un médium en entrée ou en sortie* ». Ces auteurs, cependant, ont tendance à éliminer de la catégorie multimédia les logiciels ne contenant que du texte et des images: « *après tout, les journaux avec du texte et des images ne sont pas considérés comme des publications multimédia* ». [BLA92b]

C'est à cette dernière définition [BLA92b] que nous nous en tiendrons dans la suite du travail.

Il ne faut pas confondre logiciel multimédia et logiciel hypermédia. Lorsqu'on dit qu'un logiciel est multimédia, cela signifie qu'il utilise différents média pour présenter l'information mais cela ne dit rien sur la façon dont l'information est organisée et peut être consultée. Par contre, lorsqu'un logiciel est hypermédia, cela implique une certaine organisation pour présenter cette information.

Un logiciel hypermédia est un logiciel à la fois multimédia et hypertexte. En ce qui concerne les formes de l'information accessible par le lecteur, il n'y a pas de différence entre un logiciel hypermédia et multimédia: dans les deux cas, l'information peut être présentée sous forme de texte, photos, vidéo... La différence réside dans l'organisation de l'information. En effet, dans un logiciel hypermédia, l'organisation est la même que dans un logiciel hypertexte.

Cela revient à dire que les informations sont réparties en éléments ou noeuds. Mais ces éléments, au lieu d'être attachés les uns aux autres comme les wagons d'un train, sont marqués par des liens sémantiques, qui permettent de passer de l'un à l'autre lorsque l'utilisateur les active. Les liens sont physiquement ancrés à des zones, par exemple à un mot ou une phrase.

Le procédé hypertexte permet au lecteur, en réponse à ses demandes, de constituer progressivement à l'écran un assemblage fugace d'éléments. Le lecteur se déplace en désignant à l'aide d'un pointeur le lien activable qui déclenche l'affichage du ou des éléments souhaités. L'existence du lien lui est montrée par une caractéristique physique de la zone qui le contient. [LAU92]

b. Caractéristiques

Quelle que soit la définition que l'on retient, chacune s'accorde à reconnaître que les deux caractéristiques principales des logiciels multimédia sont: [GRE93]

- leur dynamisme;
- leur interactivité.

Dire qu'un logiciel est dynamique, cela signifie qu'elle est basée sur le temps; elle peut l'être de deux manières. Premièrement, si plusieurs média sont utilisés simultanément, ceux-ci doivent

être intégrés de façon à respecter des contraintes temporelles (par exemple, l'emploi d'une séquence vidéo et d'une bande sonore exige une certaine synchronisation). Deuxièmement, dans un logiciel "time-based", l'information peut évoluer et être différente d'un instant à l'autre (exemple: le son). Ce dynamisme différencie les logiciels multimédia des logiciels graphiques dans lesquels le temps ne joue pas (le texte et les photos sont des média figés qui n'évoluent pas).

La caractéristique d'interactivité est moins intrinsèquement liée aux logiciels multimédia. En effet un logiciel multimédia n'est pas, par nature, interactif: il existe des présentations où l'utilisateur suit passivement ce qui se déroule à l'écran. Cependant, quand on analyse les logiciels multimédia, on s'aperçoit qu'ils sont majoritairement interactifs. On dira donc que le dynamisme est une propriété innée des logiciels multimédia alors que l'interactivité est une propriété acquise.

c. Le marché des logiciels multimédia

Si les bornes interactives sont un type de logiciels multimédia, ceux-ci ont trouvé d'autres marchés tels que les logiciels éducatifs, les logiciels d'animation, les encyclopédies, les jeux ou encore les logiciels roses. Tous ces types de logiciels ont tiré le meilleur de ce que pouvait leur apporter le multimédia.

Les logiciels éducatifs ont toujours existé mais l'apparition du multimédia a favorisé leur essor en les rendant moins rébarbatifs. Les encyclopédies en intégrant les différents supports d'information (texte, son, photos et vidéo) constituent une source de renseignements unique. Avant de pouvoir utiliser les images, le son et la vidéo, les logiciels d'animation et les jeux étaient beaucoup plus limités. Dans ces types de logiciels, l'important est d'avoir les meilleurs graphismes. Les fournisseurs se livrent donc une guerre dans laquelle chaque bataille est gagnée par celui qui a le plus de sons, d'images et de vidéos à un moment donné. Quant aux logiciels roses, ils connaissent une évolution incroyable: au début, il ne s'agissait que de personnes grossièrement dessinées et maintenant, il s'agit de véritables films tournés uniquement pour être mis sur CD-ROM.

Puisque le sujet de ce mémoire est l'élaboration d'une borne interactive multimédia, nous avons consacré ce chapitre à l'étude théorique des bornes interactives et du multimédia. Les chapitres suivants seront axés sur la réalisation proprement-dite de la borne. Mais, avant de pouvoir commencer sa construction, il faut connaître avec quelques précisions les besoins du client c'est-à-dire rédiger une première esquisse du cahier des charges. Cette esquisse est présentée au chapitre 3.

3. Le cahier des charges de Télémaque

Avant de démarrer la réalisation d'un logiciel, il faut rencontrer le client afin d'élaborer le cahier des charges du logiciel. Ce que doit contenir ce document est explicité à la section 3.1; les concepts ainsi dégagés seront appliqués à notre problème dans la section 3.2.

3.1. La conception standard d'un cahier des charges

Avec le progrès de la technologie, la gestion de la qualité des logiciels devient essentielle. Un des moyens d'établir un système de gestion de la qualité est de fournir des conseils pour l'assurance de la qualité du logiciel. C'est dans cette optique qu'a été créée la norme ISO 9001 qui traite des *normes pour la gestion de la qualité et l'assurance de la qualité* et qui est une spécialisation de ISO 9000.

Les lignes directrices de ISO 9001 concernent les contrats entre deux parties pour le développement d'un logiciel. Elles ont pour but de décrire les moyens de gestion et les méthodes préconisées pour réaliser un logiciel qui réponde aux besoins du client.

Cette section explique de manière précise ce que doit contenir le cahier des charges, en se basant sur [ISO91]; certaines définitions sont empruntées à Mylopoulos [MYL95].

3.1.1. Spécifications des besoins du client

La spécification des besoins du client est un document qui regroupe les exigences du client. Dans certains cas, ce document est fourni directement par le client et, dans d'autres cas, il est établi par le fournisseur en étroite collaboration avec le client. Il est important que le client valide le document car celui-ci, en plus de servir d'énoncé du problème, de documentation et de base pour la réalisation du logiciel, peut être utilisé comme contrat.

Les exigences du client peuvent être de deux types: fonctionnelles et non-fonctionnelles. Les exigences fonctionnelles sont « *une description de ce que le système doit faire* » et les exigences non-fonctionnelles sont « *des contraintes globales disant comment le système doit être construit et comment il doit fonctionner* » [MYL95].

Parmi les contraintes non-fonctionnelles les plus importantes, on distinguera

- Le taux de panne

Cette contrainte, également appelée fiabilité, est « *la capacité du logiciel à se comporter constamment de manière acceptable pour l'utilisateur lorsqu'il fonctionne dans l'environnement pour lequel il a été créé* » [MYL95]. Par exemple, le client peut exprimer

une contrainte de fiabilité de la manière suivante: le logiciel ne doit pas avoir plus de x erreurs par 1000 lignes de code. Le taux de panne concerne également les pannes du matériel.

- La tolérance aux fautes
Il s'agit de la capacité de "récupération" du logiciel après une panne. Par exemple, lorsqu'une erreur se produit, le logiciel peut-il continuer à fonctionner ou non ?
- La possibilité de récupération
Lorsque cette contrainte est exprimée, le logiciel doit effectuer des sauvegardes plus ou moins régulières.
- La facilité d'utilisation
Cette facilité d'utilisation du logiciel concerne l'apprentissage du logiciel par ses utilisateurs et son exploitation par du personnel désigné à cet effet.
- La performance
Le client peut introduire des contraintes de performance au niveau du temps de réponse ou au niveau des ressources utilisées.
- La facilité de maintenance
Pour faciliter la maintenance du logiciel, le programmeur peut se voir imposer des obligations de documentation, de programmation structurée et de standards d'analyse.
- La portabilité
Cette contrainte est exprimée lorsque le client prévoit que le logiciel devra pouvoir être exécuté dans différents environnements.
- La sécurité
Cette contrainte exprime le niveau des mesures à prendre pour protéger le logiciel face à des utilisations malveillantes.

3.1.2. Plan de développement

Le plan de développement doit définir la méthodologie qui va être utilisée pour transformer les spécifications des besoins du client en un logiciel. Il doit identifier les différentes phases qui vont être nécessaires à cet effet, les éléments d'entrée et de sortie de chaque phase et les procédures de vérification à effectuer à la fin de chaque phase.

3.1.3. Plan qualité

Le plan qualité doit spécifier des points tels que les objectifs qualité (exprimés en termes mesurables chaque fois que cela est possible), les critères définis d'entrée et de sortie de chaque phase de développement et l'identification des types de tests à mener.

3.2. Application à notre borne interactive

Notre cahier des charges comprend les spécifications des besoins du client mais ni de plan de développement ni de plan qualité. En effet, aucune méthodologie stable n'existant encore pour le développement d'un logiciel multimédia, ce sera un des objectifs de ce travail d'en proposer une. D'où l'absence des plans de développement et de qualité.

Le langage utilisé pour décrire les besoins du client est le langage naturel. Si nous avons préféré celui-ci à d'autres langages plus formels malgré ses défauts bien connus, c'est parce qu'il facilitait le dialogue avec le client et que, vu la nature du logiciel, nous avons décidé d'impliquer celui-ci le plus tôt possible dans la réalisation du projet. Nous allons diviser les besoins en deux parties: ceux qui engendrent des contraintes fonctionnelles et ceux qui donnent lieu à des contraintes non-fonctionnelles.

3.2.1. Les contraintes fonctionnelles

Les besoins exprimés par le client (l'école de commerce de Strasbourg) n'étant pas très précis, nous avons dû nous-mêmes prendre de nombreuses initiatives. Nous avons quand même pu affiner ces contraintes non-fonctionnelles progressivement, notamment par l'utilisation d'une méthodologie adéquate.

Au point de départ, c'est à dire lors de notre première rencontre, les contraintes du client s'établissaient ainsi.

Le logiciel devra

- contenir des informations sur Strasbourg et sa cinquantaine d'universités partenaires¹, informations qui devront être en 5 langues (français, anglais, allemand, espagnol, italien);
- avoir recours aux média texte, son et graphique;
- présenter la ville et l'université partenaire;

¹ Par universités partenaires, nous entendons les universités avec lesquelles l'IECS a des contacts au travers de réseaux tels que Erasmus et Comett.

- présenter des informations pratiques concernant le calendrier académique, la culture et les loisirs, les interlocuteurs privilégiés, la vie associative des étudiants, le sport, les banques, les postes...
- permettre d'imprimer les informations consultées tout en évitant qu'un même utilisateur ne puisse imprimer un trop grand nombre de copies;
- avoir recours à la vidéo pour présenter Strasbourg et l'IECS;
- permettre le changement de langue à tout moment;
- pouvoir être alimenté en informations c'est-à-dire que toute nouvelle université pourra être ajoutée à la liste.

3.2.2. Les contraintes non-fonctionnelles

Nous allons passer en revue les principales contraintes non-fonctionnelles qui s'appliquent à la conception de notre logiciel, qu'elles aient ou non été exprimées par le client.

a. Taux de panne

Si elle n'a pas été exprimée directement par le client, cette contrainte nous semble primordiale. En effet, étant donné que le logiciel est exposé au grand public, il serait très gênant qu'apparaissent régulièrement des erreurs nécessitant un redémarrage complet (une personne ne peut pas rester constamment à côté de l'ordinateur pour effectuer cette fonction). De plus, entre le moment où la faute apparaît et le moment de la remise en route, la borne serait inutilisable. Tous ces ennuis risqueraient de décourager les étudiants. Nous allons donc essayer de réduire ce taux autant que possible sachant que les pannes hardware ne sont pas de notre ressort.

b. Possibilité de récupération

Il n'y a pas ici de réelle nécessité de sauvegardes régulières. La seule chose à faire est d'effectuer une copie de la base de données après chaque mise-à-jour. Cette tâche incombe au personnel d'exploitation.

c. Facilité d'utilisation

Cette contrainte est essentielle. En effet, le logiciel doit être facile à utiliser car les étudiants ne sont pas nécessairement familiers avec l'informatique et, surtout, il n'y a pas de possibilité de formation: la borne étant exposée au grand public, tout le monde peut l'utiliser et le nombre d'utilisations varie entre une fois et quelques fois. Il faut donc que la compréhension du logiciel soit optimale dès la première fois. Au niveau de l'exploitation (mise-à-jour de la base de données), elle doit être aussi simple que possible car le personnel est, une fois de plus, novice

en informatique. Mais au contraire des utilisateurs, une formation du personnel d'exploitation est ici envisageable.

d. Performances

Si rien n'a été exigé au niveau temps et ressources, nous allons évidemment tenter d'atteindre des chiffres raisonnables. Un temps de réponse trop long pour ce type de logiciel découragerait les futurs utilisateurs et un espace disque requis trop important risquerait de poser des problèmes au client. Notre tâche sera de déterminer à partir de quel moment les chiffres pourront être qualifiés de raisonnables.

e. Maintenance

Afin de faciliter la maintenance de notre logiciel, nous allons utiliser une méthodologie précise, effectuer une architecture, adopter une programmation structurée et apporter autant de documentation que possible.

f. Portabilité

L'environnement P.C. est imposé par le client. Par contre, aucune contrainte de portabilité n'a été exprimée.

g. Sécurité

Les informations contenues dans la borne sont accessibles par tout le monde. Comme c'est un logiciel public, il serait inconcevable d'en limiter l'accès à l'aide d'un processus d'identification ("user name") et d'authentification (mot de passe).

Par contre, il faut protéger l'accès à la base de données pour éviter la transformation ou la destruction d'informations. C'est pourquoi un processus d'identification et d'authentification est nécessaire à ce niveau.

Ce chapitre avait pour but de définir les attentes du client, sans lesquelles il est impossible de lancer le processus de conception du logiciel. Nous pouvons maintenant, dans le chapitre 4, définir la méthodologie que nous allons utiliser pour obtenir un produit fini conforme à la demande du client.

4. La méthodologie

Nous nous trouvons ici face à un choix important: celui de la méthodologie à adopter. Dans le domaine du multimédia, ce point n'a pas encore fait l'objet de beaucoup de recherches vu la nouveauté de la technologie. En effet, lors de chaque progrès technologique, les personnes résolvent, dans un premier temps, les problèmes techniques qui surviennent, puis, dans un deuxième temps, prennent du recul pour analyser la situation de façon plus méthodologique. Le multimédia en est l'illustration: la technique commençant seulement à être maîtrisée, les études méthodologiques le concernant ne sont pas encore très répandues.

Conscients de la nécessité d'une méthodologie, nous sommes partis de la question suivante: pourquoi le multimédia aurait-il besoin de méthodologies différentes des logiciels "traditionnels". Les programmes multimédia ne diffèrent en effet des autres que par l'utilisation de plusieurs média au lieu d'un seul. Pourquoi, dès lors qu'on utilise de la vidéo plutôt que du texte, faudrait-il révolutionner le domaine des méthodologies. Nous ne voyons donc pas, au niveau du multimédia, ce qui nous empêcherait d'utiliser ce qui existe déjà. Par contre, le fait que le logiciel à construire soit une borne interactive va avoir une influence primordiale dans le choix de notre cadre méthodologique.

Dans ce chapitre, nous allons, dans un premier point, montrer les forces et faiblesses des méthodologies existantes (section 4.1). Ensuite, nous présenterons la méthodologie utilisée pour le développement de la borne Télémaque (section 4.2) et la justifierons dans un dernier point (section 4.3).

4.1. Etude des méthodologies existantes

Nous allons rapidement parcourir quelques méthodologies existantes [AGR86] et expliquer les raisons pour lesquelles nous ne les avons pas retenues.

4.1.1. Modèle de la cascade

Le modèle de la cascade est présenté à la Figure 4-1.

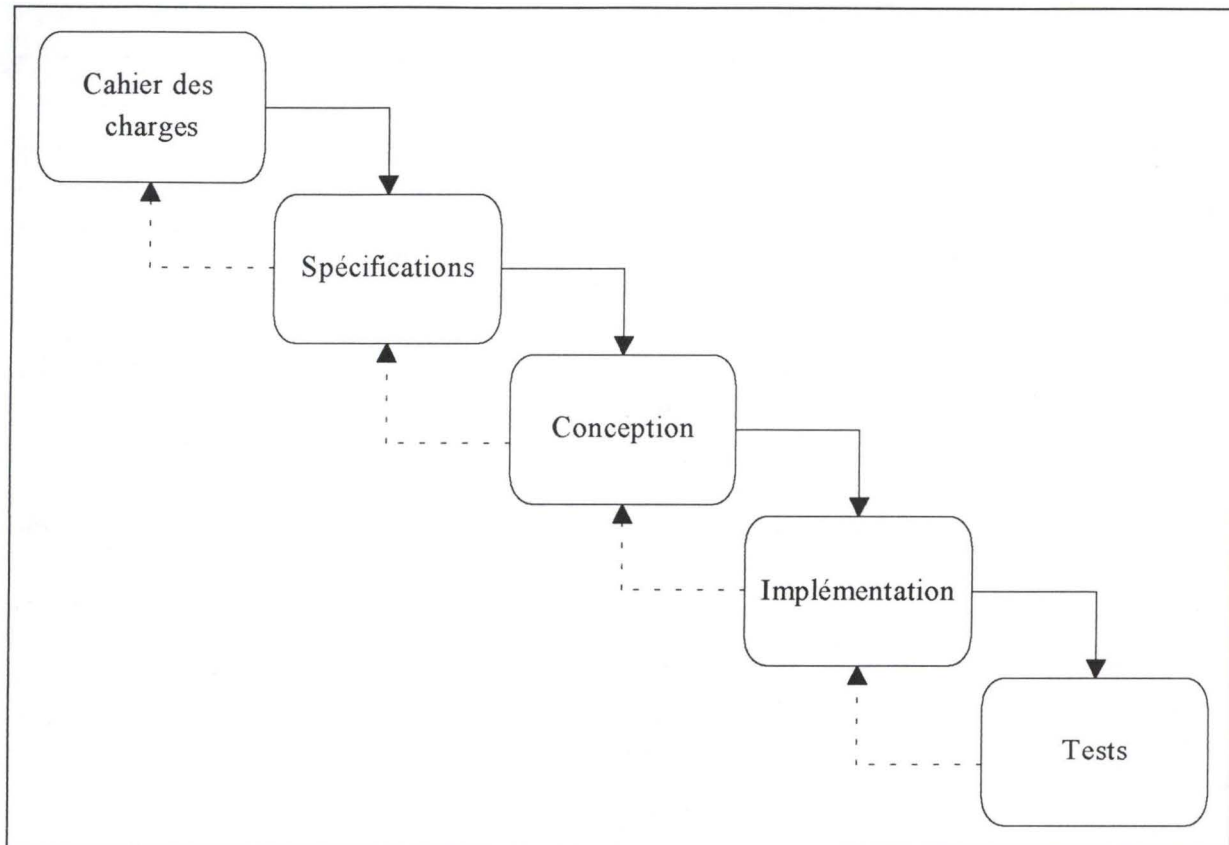


Figure 4-1. Modèle de la cascade

Ce modèle ne demande pas beaucoup d'explications: à partir du **cahier des charges**, l'analyste établit les **spécifications** du logiciel desquelles il va dériver la **conception** logique. Il suffit alors d'**implémenter** cette conception dans un langage de programmation pour obtenir le système exécutable. C'est sur ce système que sont opérés les **tests**.

Evaluation

Ce modèle ne permet pas aisément une validation rapide par le client, il faut bien souvent attendre la fin de l'implémentation pour que ce dernier soit tout à fait apte à donner son avis. S'il souhaite apporter alors des changements, ceux-ci peuvent avoir des répercussions jusque sur les spécifications. Or, dans notre cas, une participation du client est nécessaire dès le départ pour préciser les besoins exprimés au chapitre précédent

4.1.2. L'approche par prototypage

L'approche par prototypage est présentée à la Figure 4-2.

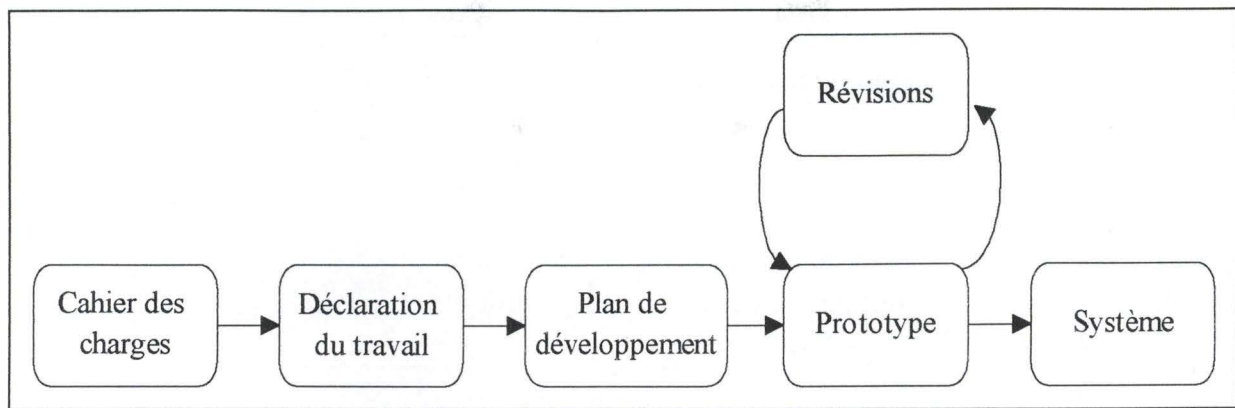


Figure 4-2. L'approche par prototypage

La **déclaration du travail** sert à préciser le **cahier des charges** (certaines choses sont difficiles à exprimer par les moyens conventionnels). C'est à ce moment qu'il convient de préciser les objectifs du prototype. Une fois les objectifs fixés, on peut élaborer le **plan de développement** c'est à dire déterminer l'argent et le temps qui vont être consacrés à la réalisation du prototype.

Une validation rapide par le client est le but de cette approche. Pour cela, il convient d'utiliser un langage de haut niveau pour accélérer le processus. Après cette analyse par le client, certaines **révisions** sont apportées au **prototype**. Une fois que celui-ci est accepté, il reste à le faire évoluer vers le **système** final.

Evaluation

Cette approche est conseillée lorsque le client ne sait pas très bien ce qu'il veut (« I'll know what I want when I see it ») et lorsque le rôle joué par les interfaces est primordial. Ces deux considérations correspondent tout à fait à notre situation. Malheureusement, le développement d'un réel prototype (qui pourrait évoluer vers un système opérationnel) coûterait trop de temps par rapport au temps que réclame la mise au point d'une borne interactive telle que Télémaque et nous empêcherait sans doute de mener ce projet à son terme. Cependant, les idées qui sous-tendent cette approche nous paraissant intéressantes, nous les reprendrons donc dans la méthodologie que nous adopterons.

4.1.3. L'approche opérationnelle

L'approche opérationnelle est présentée à la Figure 4-3.

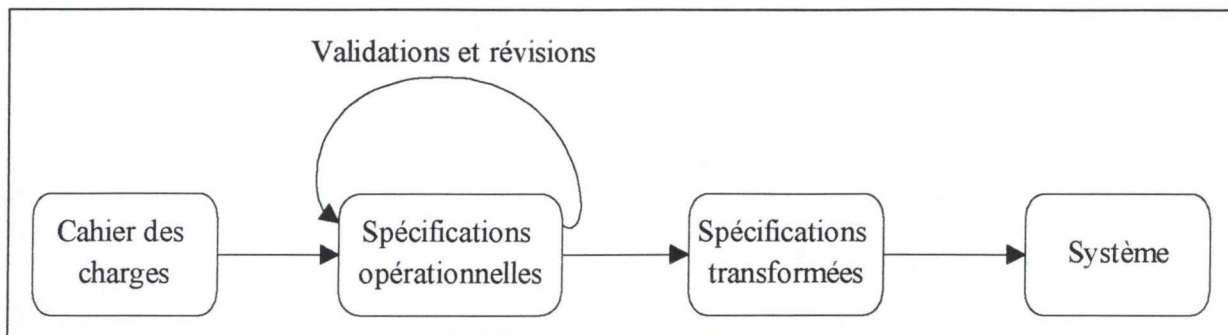


Figure 4-3. L'approche opérationnelle

L'étape de conception de la **spécification opérationnelle** représente en fait la fusion entre deux étapes du modèle de la cascade: l'élaboration des spécifications fonctionnelles et la conception logique. Le "what" et le "how" ne font donc plus qu'un. Le but recherché est d'avoir, le plus tôt possible, les comportements simulés du futur système afin que ceux-ci puissent être examinés par le client. Ceci ressemble à ce que nous avons vu dans le point précédent, les spécifications opérationnelles pouvant être considérées comme une sorte de prototype.

Une fois ces spécifications validées, le développeur **transforme** les mécanismes de comportement produits en structures qui répondent aux réalités de l'environnement-cible.

Evaluation

Cette approche semble intéressante même si nous avons toujours séparé clairement, jusqu'à présent, l'énoncé du problème (le "what") et une de ses solutions logiques (le "how"). Comme pour l'approche par prototypage, nous n'appliquerons pas cette méthode telle quelle car elle est encore au stade expérimentale.

4.1.4. L'approche mixte

L'approche mixte est présentée à la Figure 4-4.

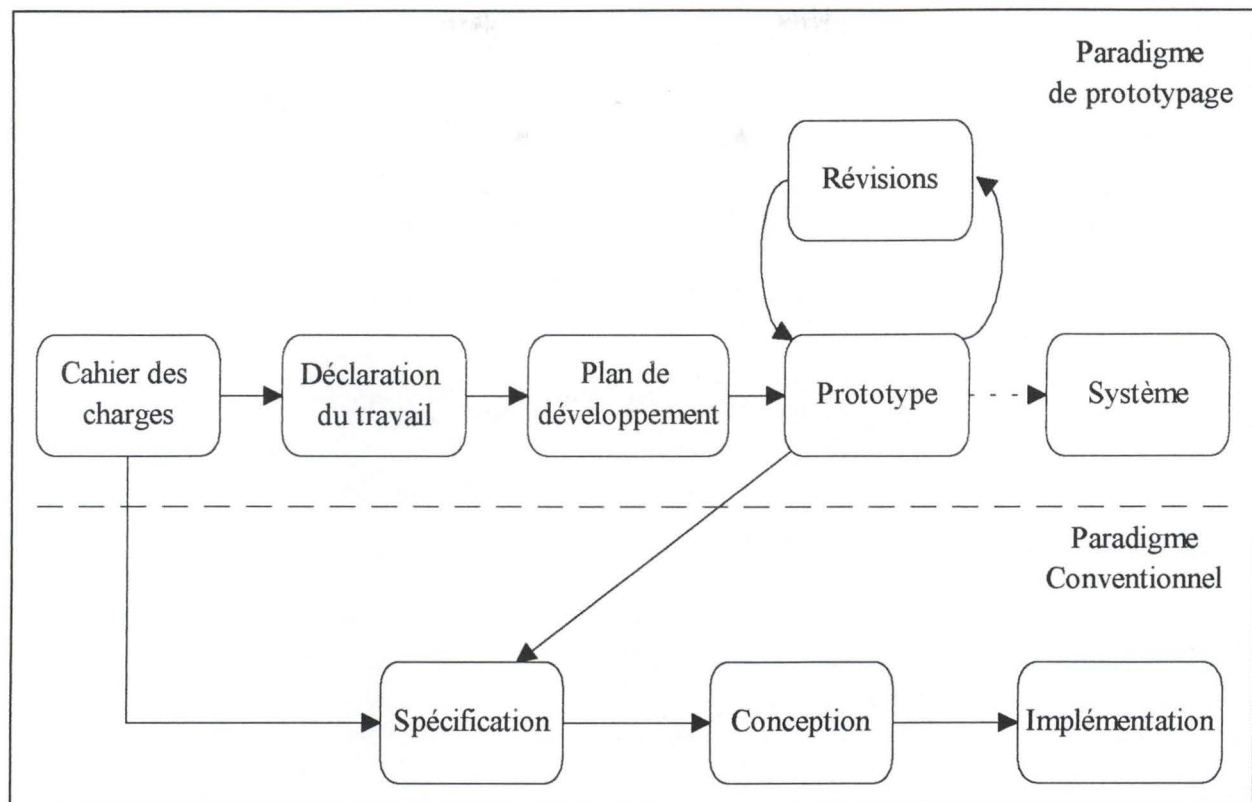


Figure 4-4. L'approche mixte

Cette approche commence comme le prototypage et se termine de façon plus conventionnelle. **Déclaration du travail, plan de développement, prototype, révisions** et enfin **prototype accepté** sont les premières étapes, identiques à l'approche par prototypage. C'est alors que le changement se produit: au lieu de faire évoluer le prototype vers un système final, on se sert de ce prototype accepté pour définir de façon très précise les besoins du client (cahier des charges). On obtient ainsi d'excellentes **spécifications** et on peut, muni de celles-ci, reprendre un développement plus conventionnel: phases de **conception** puis d'**implémentation** (système final).

Evaluation

Cette approche nous paraît intéressante car nous nous trouvons dans un environnement propice à son application. Comme nous l'avons déjà dit, les besoins du client sont assez vagues et les interfaces jouent un rôle primordial dans le produit final. L'élaboration d'un prototype nous permettrait de résoudre ces deux problèmes (besoins précisés et interfaces validées très tôt). Il nous resterait alors à reprendre un développement plus traditionnel. Nous allons donc nous baser fortement sur cette approche en la modifiant quelque peu pour répondre à certaines de nos exigences (le développement d'un prototype opérationnel nous prendrait trop de temps).

4.2. La méthodologie utilisée pour Télémaque

La méthodologie retenue est une méthodologie assez empirique se basant à la fois sur l'approche mixte que nous venons de présenter et sur trois grands principes [GOU87] :

- identifier les futurs utilisateurs en analysant ce qu'ils devront faire avec la borne et en les impliquant dans le développement;
- réaliser des expériences empiriques, des tests tout au long du développement;
- travailler selon un mode itératif c'est-à-dire qu'un problème découvert lors d'un test utilisateur est immédiatement résolu (voir Figure 4-5).

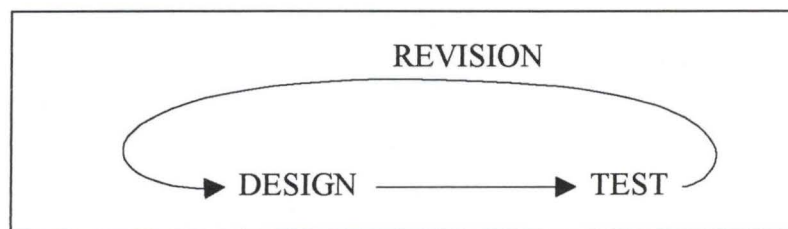


Figure 4-5. Design itératif

Le respect de ces trois principes augmente la charge de travail au départ mais diminue certains risques et donc réduit cette charge de travail par la suite.

Suivre ces trois principes est une condition nécessaire de succès mais ce n'est pas une condition suffisante. Il existe d'autres facteurs de réussite: [GOU87]

- les développeurs doivent être impliqués dans toutes les phases de construction du produit; ils doivent être motivés et doivent avoir une bonne connaissance du domaine dans lequel ils travaillent;
- les développeurs doivent avoir les bonnes personnes et le bon matériel sous la main;
- le client doit contribuer au projet.

La suite de cette section est consacrée à la présentation de notre méthodologie (voir Figure 4-6).

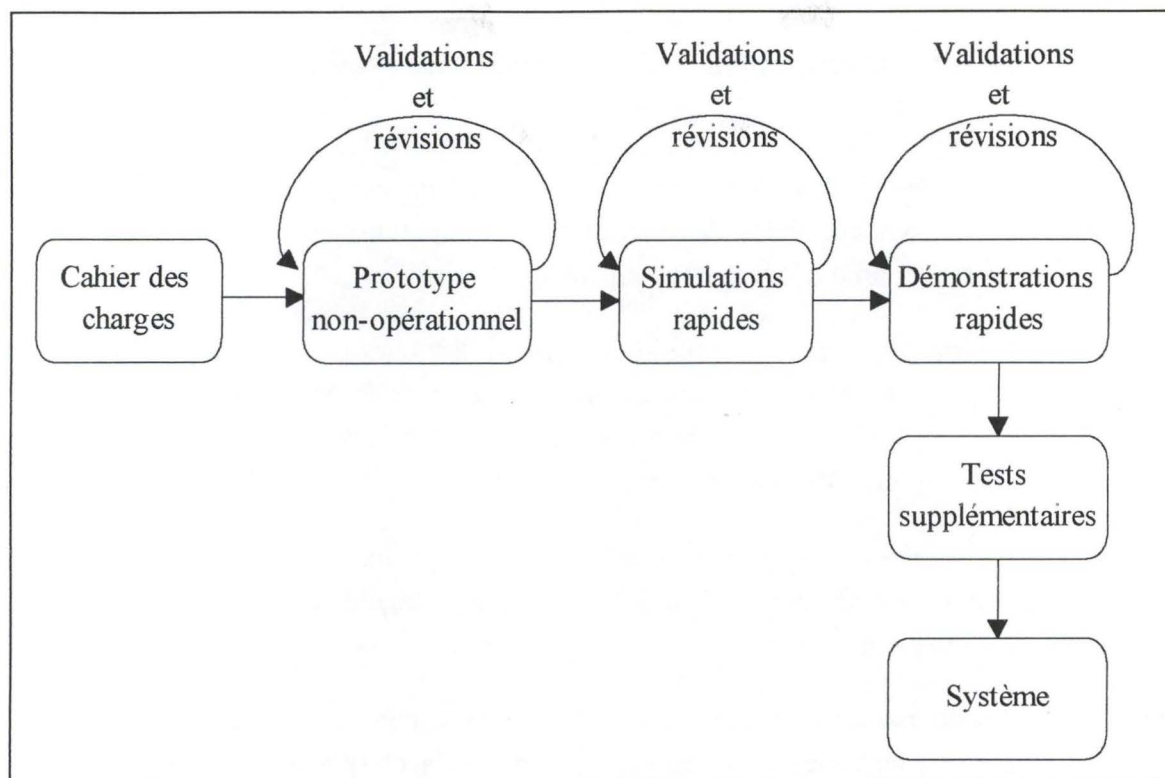


Figure 4-6. La méthodologie retenue

4.2.1. Le prototype non-opérationnel

L'élaboration du prototype non-opérationnel est matérialisée par la conception de la séquence d'écrans que l'utilisateur aura sous les yeux lorsqu'il consultera la borne. Le client voit ainsi à quoi ressembleront les écrans ainsi que les transitions entre ceux-ci. Il peut aussi valider les types d'informations contenues dans la borne en regardant les titres des items de menu.

Cette étape permet donc une première réaction critique au produit par le client et les futurs utilisateurs, ce qui peut entraîner des modifications dans le prototype sans que la moindre ligne de code ne soit écrite.

Le but de cette étape est de préciser les besoins du client. La grosse différence avec l'élaboration du prototype dans l'approche mixte réside dans le fait que notre prototype n'est pas opérationnel et ne peut donc pas évoluer vers un système final comme dans l'approche par prototypage.

Lorsque le prototype est avalisé par les développeurs et les utilisateurs, on peut reprendre le développement en suivant une méthodologie plus traditionnelle avec des phases de conception et d'implémentation, même si la phase de conception des interfaces a déjà été entamée.

4.2.2. Simulations rapides: développement des interfaces

En utilisant la séquence d'écrans réalisés à l'étape précédente comme base de la programmation, on peut commencer à programmer. Il est important de choisir un langage d'implémentation qui donne la possibilité de rendre les écrans opérationnels sans devoir écrire de lignes de code concernant un quelconque traitement.

Cette étape est également suivie de tests utilisateurs. Ces utilisateurs peuvent désormais voir les écrans implémentés et la réaction du programme ce qui n'était pas possible à l'étape précédente. Ici encore, les modifications sont aisées car tout le code n'est pas écrit (on peut même changer un écran "on-line" devant l'utilisateur).

Une première analyse du système d'aide peut être effectuée: on met l'utilisateur devant la borne et on analyse sa réaction. Quand il est en difficulté, on lui demande ce qu'il voudrait voir ou entendre pour être capable de continuer.

Quand on parle de développement des interfaces, il s'agit non seulement de la conception des écrans mais également de l'élaboration des autres types d'interfaces (par exemple, l'interface sonore).

4.2.3. Démonstrations rapides: développement de la base de données

Cette étape a lieu lorsque le nombre de fonctions ou le contenu du programme a augmenté et est devenu suffisant pour réaliser des tests sur une plus grande échelle. Dans notre cas, le nombre de fonctions va augmenter avec le développement de la base de données.

Les tests utilisateurs effectués après cette étape servent à vérifier la correction des requêtes vers la base de données ainsi que le temps de réponse du programme modifié (il va maintenant chercher les informations dans la base de données). L'analyse du système d'aide peut évidemment être poursuivie.

4.2.4. Tests supplémentaires

Lorsqu'on a terminé le programme selon les spécifications et que les différents tests ont été réalisés, il faut encore et toujours procéder à des tests utilisateurs pour éliminer les erreurs résiduelles et être sûrs que la borne répond bien aux attentes des utilisateurs.

4.3. Justifications de la méthodologie choisie

Dans une borne interactive, les interfaces ont plus d'importance et sont plus complexes que les algorithmes de traitement; c'est pourquoi nous avons placé les interfaces au centre de la méthodologie. Nous avons commencé par l'écriture d'un prototype et la conception des interfaces plutôt que par établir une liste des fonctions du logiciel car cela semblait difficile au

départ vu que les spécifications étaient très vagues. Il était donc important que le client puisse valider ce que nous faisons. Cette tâche était facilitée par l'optique choisie.

En effet, grâce à cette méthodologie, il est possible d'identifier facilement et très tôt les directions à suivre ou à ne pas suivre et de procéder aux modifications nécessaires.

De plus, cette méthodologie a déjà fait ses preuves puisqu'elle a été utilisée pour développer la borne d'envoi de messages aux jeux olympiques de Los Angeles en 1984 [GOU87]. Cette borne permettait aux athlètes et à leur famille de correspondre pendant toute la durée des jeux. Les messages étaient envoyés par téléphone et enregistrés sur un ordinateur. Les athlètes, par exemple, utilisaient une des bornes pour écouter les messages qu'ils avaient reçus.

Cette borne, qui a été un véritable succès, possède beaucoup de points communs avec la nôtre. Elle

- est utilisable par tout le monde (en majorité par des utilisateurs novices);
- est sujette au sabotage et aux abus;
- est multimédia: le son est utilisé pour restituer un message et la vidéo est utilisée pour expliquer comment marche la borne;
- est multilingue: des traductions ont dû être faites et les athlètes devaient pouvoir changer de langue à tout moment;
- a des interfaces de haute qualité.

La seule différence dans la méthodologie concerne la rédaction des manuels d'utilisation. Pour la borne des J.O., ceux-ci ont été rédigés très tôt (juste après l'élaboration du scénario). Cela se justifiait par le fait qu'une utilisation de la borne coûtait assez cher car il s'agissait d'une communication téléphonique internationale. Il était donc important que les familles maîtrisent le logiciel avant de l'employer. Dans notre cas, nous n'avons pas rédigé de manuel d'utilisation du logiciel car notre but était que le logiciel se suffise à lui-même c'est-à-dire qu'il ne nécessite pas de recours à un quelconque manuel pour être utilisé correctement.

De la méthodologie que nous avons présentée dans ce chapitre, les points les plus importants sont l'élaboration du prototype et la conception des interfaces. Pour réaliser le prototype, nous avons rédigé un scénario que nous explicitons au chapitre suivant. Comme nous le montrerons, le concept de scénario correspond à la définition de prototype que nous avons donnée dans la présentation de la méthodologie, à savoir une séquence d'écrans.

5. Le scénario

Pour réaliser une borne interactive (et plus généralement un logiciel multimédia), il faut d'une façon ou d'une autre construire un scénario. Ce concept vient directement du monde du cinéma. C'est pourquoi nous allons présenter les différents aspects du scénario informatique en nous référant d'abord au scénario cinématographique (section 5.1).

Nous nous référerons ensuite au scénario publicitaire (section 5.2) qui, bien que dérivé du scénario cinématographique, va introduire de nouveaux concepts.

A l'issue de ces comparaisons, nous donnerons quelques caractéristiques propres au scénario informatique avant de présenter le scénario de notre borne (section 5.3).

5.1. Rapports entre scénario cinématographique et scénario informatique

Pour effectuer la comparaison entre les scénarios cinématographique et informatique, on recourra principalement à l'ouvrage de Chion [CHI85].

5.1.1. Première définition d'un scénario

Scénario cinématographique

Si la pratique du scénario n'a cessé de se développer au cours de ce siècle, il semble bien que la définition de cette activité reste imprécise. On peut dire que le scénario est la description d'une oeuvre narrative qui n'existe pas encore et qu'il a pour fonction de la rendre réalisable. Dans cette définition, le scénario n'existe qu'en tant qu'ossature d'un récit destiné à être développé plus tard sous une autre forme. A ce moment, le scénario sera détruit car s'il reste scénario, ce sera le signe de son échec. La métaphore de la chenille et du papillon convient parfaitement pour expliquer ce phénomène: la chenille a pour but de donner naissance au papillon même si, ce faisant, elle s'anéantit. [PEE86]

Scénario informatique

Cette première définition du scénario cinématographique semble intéressante et peut être transposée très facilement au monde informatique: *le scénario informatique est la description d'un logiciel qui n'existe pas encore et il a pour fonction de le rendre réalisable*. Ici aussi, si le scénario reste scénario, ce sera le signe de son échec.

5.1.2. Les éléments du scénario

Les principaux éléments d'un scénario sont: l'histoire et la narration, l'unité de sujet, l'intrigue secondaire et le dialogue.

a. Histoire et narration

Scénario cinématographique

Il faut faire en matière de cinéma une distinction capitale entre l'histoire et la narration.

L'histoire, c'est ce qui se passe quand on met le scénario à plat dans l'ordre chronologique.

La narration (ou récit, discours, construction dramatique, etc.), c'est la façon dont l'histoire est racontée. Par exemple, c'est la façon dont les événements et les données de l'histoire sont portés à la connaissance du public. Cet art de la narration peut à lui seul donner de l'intérêt à une histoire sans surprise ou, au contraire, gâcher une bonne histoire.

Une histoire est racontable par différents moyens (roman, radio, film, pièce de théâtre, etc.) alors que la narration est spécifique à chaque moyen employé. En effet, l'art de raconter une même histoire ne sera pas identique dans un roman ou dans un film.

Un scénario raconte donc une histoire selon une certaine manière (narration).

Scénario informatique

Dans un logiciel multimédia, l'histoire, c'est l'information qui doit être présentée à l'utilisateur. La façon dont on va organiser sa présentation représente la narration.

Tout comme une histoire est racontable par différents moyens, des informations peuvent, dans un logiciel multimédia, être présentées sous différentes formes: texte, son, photo, vidéo... L'art de présenter ces informations va également dépendre de la forme choisie.

Nous pouvons donc dire que *le scénario informatique présente de l'information à l'utilisateur selon une certaine organisation.*

b. L'unité de sujet (intrigue principale)

Scénario cinématographique

Le critère d'unité (de sujet, de forme, d'action, de ton, etc.) est un vieux critère qui nous vient du théâtre et qui a été repris au cinéma (même si certains réalisateurs cherchent à faire le contraire dans leur film). Pour rédiger un scénario, le plus important semble être l'unité de sujet: il ne faut pas diluer l'unité du scénario dans de trop nombreuses intrigues secondaires.

Le scénariste doit donc distinguer ce qui est essentiel de ce qui ne l'est pas car le besoin de meubler l'action, d'inventer des péripéties, conduit souvent à accumuler des détails qui vont faire perdre de vue le sujet principal. C'est pourquoi il doit avoir constamment à l'esprit son idée de départ. S'il ne le fait pas et s'il ne manie pas aisément les règles, procédés et trucs dramatiques, il risque d'aboutir à un scénario touffu et consistant mais sans lignes de forces. En gardant l'idée principale en tête, le scénariste peut, chaque fois qu'il enrichit l'histoire, se poser la question suivante: est-ce que cela renforce l'idée ou, au contraire, est-ce que cela brouille les pistes ?

Scénario informatique

Le scénario d'un logiciel multimédia doit respecter une unité de sujet. Le développeur du logiciel doit garder en tête l'idée de départ: à quoi et à qui servira le logiciel ? Tout comme dans le cinéma, il faut manier avec prudence les astuces dramatiques (par exemple, les effets spéciaux); dans l'informatique, il ne faut pas vouloir utiliser la technologie pour la technologie mais parce que celle-ci peut aider à rendre le logiciel facile à utiliser, attrayant... Les "effets spéciaux" qu'un développeur peut insérer dans un logiciel ne doivent pas le rendre touffu. Il doit se poser la question suivante: est-ce que cela renforce l'idée ou, au contraire, est-ce que cela brouille les pistes ? On pourrait imaginer qu'un logiciel éducatif, lorsque l'utilisateur répond mal à une question, propose une longue séquence animée au lieu de fournir immédiatement la bonne réponse. Ce genre de comportement, s'il est intéressant au niveau technologique et esthétique, peut perturber l'utilisateur et ne correspond certainement pas à l'objectif initial du logiciel.

c. L'intrigue secondaire

Scénario cinématographique

Une intrigue secondaire, c'est au cinéma un écart par rapport à l'intrigue principale. Elle est utilisée pour permettre au spectateur de souffler un peu mais aussi pour étoffer, varier, distraire et parfois détourner l'attention du public. Elle peut aussi servir à présenter l'histoire d'une autre façon que ne le fait l'intrigue principale.

Scénario informatique

Un développeur peut insérer, à côté des informations principales respectant l'idée de départ du logiciel, des informations secondaires, des détails dont le but est de distraire l'utilisateur ou de lui présenter les informations sous une autre forme (à côté des informations formelles, on peut mentionner l'une ou l'autre anecdote). Dans un logiciel multimédia, les moyens de rompre la monotonie ne manquent pas. L'important est que ces informations, si elles sont secondaires, restent malgré tout de l'information.

d. Le dialogue

Scénario cinématographique

Dans tous les types de films (même les films muets où le dialogue résidait dans les mimiques des acteurs), le dialogue joue un rôle central. Les fonctions du dialogue au cinéma sont

- de faire avancer l'histoire;
- de communiquer des informations au public;
- d'établir leurs relations les uns avec les autres;
- de révéler les conflits et l'état émotionnel des personnages;
- de commenter l'histoire.

A côté de cela, le dialogue a des caractéristiques.

- Il doit être dynamique et non statique. Pour cela, il est préférable d'éviter l'alternance mécanique des questions et des réponses. De plus, dans un jeu de questions - réponses, la réplique d'un personnage à un autre peut ne pas être une réplique mais un silence, un geste ou une absence de réaction.
- Il doit être émotionnel et non rationnel.
- Il n'est pas généralement rédigé dans un style littéraire. En effet, le but d'un scénario est d'être porté à l'écran et donc, les répliques ne seront plus lues mais écoutées par le spectateur. Le style doit être un style oral car le texte parlé ne s'absorbe pas aussi facilement que le texte écrit et le pouvoir de concentration du spectateur faiblit vite.

Scénario informatique

La grande différence réside dans le fait que, au cinéma, le spectateur subit passivement le dialogue alors que, en informatique, l'utilisateur prend part à ce dialogue qui s'installe entre lui et la machine.

On retrouve, dans le dialogue informatique, certaines fonctions du dialogue au cinéma. Le dialogue doit faire progresser l'histoire c'est-à-dire faire avancer l'utilisateur dans sa consultation. Il doit servir à communiquer des informations et à établir les liens entre les différentes informations.

Au point de vue des caractéristiques, en informatique aussi il vaut mieux éviter d'employer systématiquement le jeu des questions - réponses qui risque de lasser l'utilisateur. Ce dernier peut également répondre par absence de réaction (le logiciel propose quelque chose qui sera validé après x secondes si l'utilisateur n'a pas dit le contraire) ou par geste dans le cas d'une borne interactive avec écran tactile.

Dans un programme multimédia, ce qui est destiné à être lu et ce qui est destiné à être entendu ne doit pas être rédigé dans le même style. Le texte oral doit avoir un style plus dépouillé alors

que le texte écrit peut être plus dense car la capacité de concentration de l'utilisateur n'est pas la même dans les deux cas.

5.1.3. *Les parties du scénario*

a. L'exposition

Scénario cinématographique

L'exposition est la partie initiale du scénario dans laquelle sont exposés, au spectateur, les différents éléments et points de départ à partir desquels l'histoire qui va être racontée va pouvoir fonctionner. A ce moment, le scénariste doit faire passer de l'information (personnages principaux, cadre, situation de départ, etc.); or, au cinéma, le spectateur ne retient pas l'information: il ne retient que les sentiments.

L'exposition ne doit pas avoir l'air d'être une exposition: il ne faut pas que l'acteur ait l'air forcé de dire ce qu'il dit pour communiquer au spectateur des informations.

L'art de l'exposition consiste donc à délimiter ce que le public a besoin de savoir et faire en sorte qu'il ait envie de le savoir.

Scénario informatique

L'exposition serait le(s) premier(s) écran(s) dans le(s)quel(s) seraient expliqués, à l'utilisateur, le fonctionnement du logiciel (système d'aide) et les points de départ de son utilisation (par exemple, le menu principal). C'est normalement plus facile qu'au cinéma car un utilisateur veut trouver, dans un logiciel, de l'information et non des sentiments.

Il faut essayer de rendre cette communication d'informations aussi naturelle et attrayante que possible sinon l'utilisateur risque de la sauter, pour passer directement au logiciel lui-même en se disant que son apprentissage se fera avec l'expérience (en touchant à tout).

Comme au cinéma, il faut délimiter ce dont l'utilisateur a besoin pour utiliser correctement le programme mais sans lui en dire trop. Le multimédia fournit, de ce point de vue, différents moyens d'exciter sa curiosité (par exemple, utiliser une vidéo montrant un utilisateur devant ce même programme).

b. Le climax

Scénario cinématographique

Le climax est le point culminant d'un film (en émotion, en drame, en intensité). Il est normalement situé à la fin du film et donne une issue à l'histoire. Après le climax, il ne peut y avoir que des scènes de détente.

Il existe également des climax localisés appelés temps forts. Un temps fort se prépare plus ou moins longuement car son effet n'est pas seulement fonction de sa force, mais surtout il est dû à ce qui le précède car il est un point sur une courbe. On peut donc voir une oeuvre dramatique comme une succession de courbes et de variations d'intensité.

Scénario informatique

Le climax est le point culminant de l'utilisation d'un logiciel (par exemple, c'est le moment où l'utilisateur trouve l'information qu'il cherchait). Il donne une issue à la recherche de l'utilisateur.

c. La loi de progression continue

Scénario cinématographique

La tension dramatique doit aller en croissant pour atteindre le climax à la fin du film. C'est à ce moment que, au terme d'une montée, les événements les plus frappants et les émotions les plus fortes doivent être donnés. Un scénario est donc une série de situations de crise, chacune plus grave que la précédente, menant au climax. Pour cela, il est très important d'avoir de bonnes liaisons entre les scènes: toute scène doit avancer en répondant à des questions qui ont été posées précédemment et doit en susciter d'autres.

Scénario informatique

La loi de progression continue pourrait être interprétée comme suit: tout logiciel doit permettre à l'utilisateur de progresser en son sein pour atteindre le climax. C'est à ce moment que les événements les plus frappants doivent être délivrés. En transposant, nous pouvons dire: *un scénario est une série de situations (écrans) menant au climax (l'information recherchée)*. Il faut également de bonnes liaisons entre les écrans.

d. Scènes et séquences

Scénario cinématographique

Une scène est définie comme une section de l'histoire où un événement tout entier se produit. Elle doit être identifiée par sa place et son moment (quand et où ?). La comparaison la plus célèbre est celle du corps humain: une scène représente une cellule alors que le scénario représente le corps entier.

Une séquence, unité plus grande, est définie comme une série de scènes regroupées par une idée commune.

Scénario informatique

Un logiciel peut également être découpé: une scène serait un écran et une séquence, un regroupement d'écrans qui sont liés suivant un certain critère.

e. L'accroche

Scénario cinématographique

L'accroche est l'événement frappant, bizarre, surprenant que l'on place au début de l'histoire pour capter l'intérêt du spectateur. Une accroche ne doit pas être trop spectaculaire pour ne pas rendre la suite trop fade et permettre la loi de progression continue.

Scénario informatique

En informatique, l'accroche peut être vue de deux façons.

Soit, comme au cinéma, l'utilisateur est déjà devant le logiciel et l'accroche est la présentation (multimédia) du logiciel. Ici non plus, la présentation ne doit pas être plus spectaculaire que le logiciel lui-même pour ne pas frustrer l'utilisateur par la suite.

Soit l'utilisateur n'est pas encore devant le logiciel et l'accroche doit servir à attirer son attention. C'est le cas d'une borne interactive: placée dans un lieu public, elle voit défiler beaucoup de personnes et doit essayer de "se montrer". Dans ce cas, l'accroche doit être aussi spectaculaire que possible car ces personnes ne sont pas spécialement là pour la borne.

5.1.4. Les procédés de narration

a. Règles de la dramatisation

Scénario cinématographique

Il y a des règles qu'il faut suivre pour tenter de dramatiser les faits qu'on présente au spectateur.

- La concentration.
Il faut concentrer l'histoire pour s'adapter au degré d'information assimilable par un spectateur.
- L'émotionnalisation.
L'histoire ne doit pas être racontée de manière neutre mais de façon à créer une participation émotionnelle de la part du spectateur.
- La hiérarchisation.
Il faut insister sur ce qui est important par rapport à des détails qui resteront comme un décor ou un agrément.
- La création d'une courbe.
La narration doit suivre une courbe montante (loi de progression continue) et s'appuyer sur des temps forts.

Scénario informatique

Nous allons montrer ce que ces règles impliquent en informatique.

- La concentration.

Il faut concentrer l'information pour s'adapter à l'utilisateur. En effet, un utilisateur d'encyclopédie installé dans son fauteuil sait recevoir plus d'informations qu'un utilisateur de borne interactive qui ne fait que passer quelques minutes devant celle-ci.

- L'émotionnalisation.

Quand le cadre s'y prête, l'information ne doit pas être restituée de manière neutre. Dans le cas d'une borne interactive, le développeur peut rendre l'information "vivante".

- La hiérarchisation.

L'information importante doit être mise en évidence, doit ressortir par rapport à ce qui est là à titre anecdotique.

- La création d'une courbe.

Il faut concevoir un logiciel en y intégrant des variations d'intensité pour maintenir et relancer la curiosité de l'utilisateur. Nous voulons dire par là qu'un logiciel doit, de temps en temps, surprendre l'utilisateur sinon il court le risque de devenir monotone. Par exemple, lors d'une consultation prolongée, fournir des informations en utilisant constamment le même médium risque d'ennuyer l'utilisateur. Passer du texte au son puis à la vidéo crée des ruptures propices à la stimulation de l'intérêt.

b. Informations

Scénario cinématographique

Raconter une histoire, c'est en même temps communiquer des informations au spectateur. Le scénario est pour une grande part un art de la sélection de l'information. Le tout est de déterminer la quantité d'information à fournir au spectateur et à quel moment la fournir (on peut entretenir le suspense en gardant une information secrète). Créer de petits mystères autour de certaines scènes remobilise le spectateur.

Certaines informations peuvent être implicites: elles sont supposées comme telles sauf si l'information contraire est donnée. Là encore, une fausse piste donnée au spectateur peut avoir un effet dramatique.

L'information donnée doit être limitée mais les informations importantes doivent être répétées car une information se perd parmi les autres informations, les sentiments, l'action... Or, au cinéma, il est impossible de revenir en arrière comme lorsqu'on lit un livre. Cette répétition doit être variée pour ne pas lasser le spectateur.

Scénario informatique

Le scénario informatique est également un art de la sélection de l'information. Mais ici, l'interactivité étant possible, le scénariste ne se demande pas a priori quand il va distiller telle information: cela dépendra de l'utilisateur.

De même, la quantité d'information délivrée doit respecter les capacités d'assimilation de l'utilisateur. Cependant, il n'est pas nécessaire de répéter les choses importantes vu que l'utilisateur peut revenir en arrière comme dans un livre.

L'étude du scénario cinématographique met en évidence certaines caractéristiques utilisables lors de l'élaboration d'un scénario informatique. Nous montrerons au point 5.3.3 comment nous avons introduit ces caractéristiques dans le scénario de Télémaque. Mais auparavant, nous souhaiterions montrer les apports du monde de la publicité au concept de scénario.

5.2. Rapports entre scénario publicitaire et scénario informatique

Après avoir analysé les relations entre scénario cinématographique et scénario informatique, nous allons dans cette section analyser les rapports entre scénario publicitaire et scénario informatique.

La publicité est un moyen d'information multimédia car elle peut avoir recours à différents média: vidéo, image, texte, son.

L'étude du scénario publicitaire, basée sur le livre de Peeters [PEE86], vient compléter celle du scénario cinématographique en lui ajoutant une dimension qui nous intéresse dans le cadre de ce travail: la relation avec le client et l'utilisateur final.

Peeters analyse les différents aspects d'un scénario publicitaire en se fondant sur son cycle de vie. Il est ainsi amené à examiner successivement le rôle du scénario, l'analyse du produit et du client qui est à la base du scénario, la définition des besoins et la validation par le client.

5.2.1. Rôle du scénario

Scénario publicitaire

Le scénario ou maquette publicitaire doit être validé par le client, être une anticipation du produit final et laisser un peu de champ à la réalisation. Un peu de champ signifie que le scénario doit être suivi aussi fidèlement que possible et que des digressions ne sont permises qu'au niveau de détails presque insignifiants. Au cinéma, par contre, plus de champ est laissé au réalisateur et aux acteurs.

Scénario informatique

Le rôle du scénario multimédia est identique à celui du scénario publicitaire. Ici non plus, le développeur ne doit pas prendre trop de libertés avec le scénario établi.

5.2.2. Définition des besoins

Scénario publicitaire

Pour choisir, par exemple, la photo d'une campagne publicitaire, le client donne ses envies au publicitaire. Celles-ci sont souvent très confuses, peu précises. Le client évoque tel film et tel acteur pour tenter d'expliquer ce qu'il attend. Bien souvent, il ne sait pas exactement ce qu'il veut et il attend un premier jet du publicitaire pour affiner sa pensée.

Scénario informatique

Les spécifications d'une borne interactive sont souvent confuses, non formelles car le client ne sait pas ce qu'il veut. Le scénario doit donc servir à préciser ces spécifications en permettant au client de se positionner par rapport au logiciel en sorte qu'il dise: « ça, c'est bien », « ça, c'est pas bien », « là, ce serait mieux comme ça »...

5.2.3. Analyse du produit et du client

Scénario publicitaire

En publicité, le scénario est une nécessité et est l'aboutissement d'une étude très complexe. Cette analyse porte sur le produit à promouvoir et sur l'individu susceptible de l'acheter. Le résultat de cette étude est une image précise du produit et de la clientèle potentielle. Avant d'entamer le projet en lui-même, il convient donc de définir un certain nombre de paramètres.

Scénario informatique

L'élaboration du scénario ne peut débuter qu'après avoir pris connaissance de l'environnement, du type d'utilisateur visé et de la tâche que cet utilisateur va devoir effectuer. En informatique, cette dernière étape, qui s'appelle l'analyse de la tâche, est également un passage obligé vers le scénario. Les liens entre analyse de la tâche et scénario sont expliqués plus en détail à la section 5.3.3.

Il faut tenir compte de cette analyse pendant l'élaboration du logiciel afin de ne pas sacrifier le fonctionnel au profit de l'esthétique.

5.2.4. Validation par le client

Scénario publicitaire

Lorsqu'il faut prendre des décisions importantes et engager un budget, le client doit toujours être consulté. Lorsque celui-ci donne son accord pour un texte ou une photo, le publicitaire ne doit pas, par la suite, s'éloigner de la proposition acceptée. S'il le fait, il court le risque de voir le client lui retirer son accord.

Contrairement au scénario cinématographique, le scénario publicitaire a été tellement testé que l'on y a réduit au minimum la liberté de conception. La réalisation n'a plus qu'à le suivre fidèlement car apporter des corrections à ce stade peut coûter très cher et, si cela ne marche pas, le client peut se retourner contre l'agence pour non-respect du scénario.

Scénario informatique

Lorsque le scénario final a été accepté par le client, le développeur doit le suivre le plus fidèlement possible pour ne pas risquer que le client se rétracte.

L'étude du scénario publicitaire est intéressante car on peut en déduire une marche à suivre pour concevoir un scénario informatique. Cette marche à suivre est appliquée à notre borne à la section 5.3.3.

5.3. Le scénario informatique

Après ces analyses comparatives, nous allons donner quelques définitions et caractéristiques du scénario informatique. Nous consacrerons ensuite le reste de cette section à la présentation du scénario de Télémaque et montrerons les apports des secteurs cinématographique et publicitaire au scénario informatique.

5.3.1. Définitions

Si le multimédia est « *l'association de plusieurs monomédia avec description des liens logiques qui lient les éléments d'informations monomédia* », le scénario multimédia est « *l'organisation de l'ensemble des liens avec les différents monomédia* ». [GOR95]

Selon Loeb, le scénario « *est un système de filtrage de l'information impliquant différents types d'information, des architectures de restitution et un équipement utilisateur* » [LOE92]. Le scénario sert donc à filtrer l'information entre les sources de l'information et les utilisateurs.

Nous sommes conscients qu'il existe encore d'autres définitions de scénario informatique (voir [KAU89], [POT94], [CAR94], [HSI94]) mais celle que nous retiendrons est transposée du

monde cinématographique: « *un scénario est une série d'écrans menant à l'information recherchée* ».

5.3.2. Caractéristiques

Si nous avons déjà identifié un certain nombre de caractéristiques du concept de scénario grâce au monde du cinéma, la construction d'un scénario informatique dépend de facteurs supplémentaires [LOE92] :

- le type d'utilisateur: l'expert préférera un moyen rapide de trouver l'information voulue alors que le débutant préférera un moyen facile;
- la protection de la vie privée: le scénario ne sera pas le même si certaines données sont confidentielles ou si les informations délivrées peuvent être connues de tout le monde sans restriction;
- la durée de vie de l'information: une information peut être valable quelques minutes, quelques jours, quelques années ou quelques siècles mais, en règle générale, il faut que le temps de réponse du programme soit largement inférieur à la durée de vie d'une information (un temps de réponse de l'ordre de la minute est tout à fait insuffisant pour des données qui peuvent évoluer à chaque seconde);
- les modèles de disponibilité de l'information: selon leur valeur, certaines informations doivent être plus facilement accessibles que d'autres (analyser le type de l'information et sa fréquence de consultation);
- les modèles de restitution: la restitution peut être continue (l'information est délivrée dès qu'elle est disponible) ou dépendante des requêtes de l'utilisateur;
- les modèles d'utilisation: on analyse la durée et la fréquence des sessions¹ et, si celles-ci ne sont pas uniques, il apparaît intéressant de conserver, dans le système, un historique des préférences des utilisateurs (pour lui délivrer le plus rapidement possible l'information qui l'intéresse);
- le mécanisme de feed-back: le feed-back doit-il être immédiat ou peut-il attendre jusqu'à la prochaine session ?
- la nature des média: les formes de l'information influencent les performances du scénario et des filtres qu'il utilise pour retrouver les données demandées;

¹ Une session est le temps pendant lequel un utilisateur se sert d'un logiciel.

- l'équipement utilisateur: l'équipement dont dispose l'utilisateur va influencer le scénario (le filtrage de l'information);
- les attributs du contenu de l'information: le filtre utilisé par le scénario va examiner les données (ou un descripteur de données) pour voir si elles sont pertinentes pour l'utilisateur.

Ce dernier point est particulièrement important dans les logiciels multimédia. Imaginons d'abord que l'utilisateur introduit un nom de chapitre et demande à l'ordinateur de retrouver le livre duquel il est extrait. Ceci ne pose pas de problème: l'ordinateur consulte sa base de données et compare du texte avec du texte. Imaginons maintenant que l'utilisateur introduise une composition musicale et demande de quel film elle est extraite, la réaction que doit avoir l'ordinateur est moins évidente (quel algorithme utiliser ?).

5.3.3. Le scénario de Télémaque

Nous commençons cette section par expliquer les apports du monde publicitaire car ceux-ci concernent la marche à suivre pour concevoir un scénario. De là, nous déduirons qu'avant de réaliser un scénario, il est nécessaire de rédiger une analyse de la tâche. Nous pourrons ensuite présenter le scénario de Télémaque en ayant recours à l'analyse de la tâche et au monde cinématographique. Nous terminerons cette section par quelques caractéristiques supplémentaires.

a. Apports du monde publicitaire

Avant de réaliser un scénario publicitaire (voir 5.2), il faut définir les besoins du client et analyser le produit et le client. Lorsque le scénario est terminé, il doit être validé par ce même client. On peut appliquer cette démarche à la construction d'un scénario informatique. Dans notre cas, les besoins du client ont déjà été définis dans le cahier des charges. Il ne nous reste plus qu'à procéder à une analyse de la tâche avant de pouvoir démarrer la conception du scénario.

b. L'analyse de la tâche

Avant de construire le scénario de Télémaque, nous avons donc réalisé une analyse de la tâche. Celle-ci nous a permis de mettre en évidence les différentes étapes d'une recherche "classique" d'informations. Cette découpe en étapes constitue la base de notre scénario.

La consultation d'informations n'est pas une tâche triviale à analyser, même si elle nous est tout à fait familière. En effet, lorsque nous la regardons de plus près, nous nous rendons compte qu'elle peut prendre mille et une formes. Voici deux exemples tout à fait différents de recherche d'informations.

1. Supposons qu'on désire obtenir la définition exacte d'un mot. Il suffit de saisir le dictionnaire et de procéder à une recherche de type dichotomique pour rapidement tomber sur le mot recherché et pouvoir consulter sa définition.
2. Supposons maintenant que l'on désire partir en vacances et que l'on ait plusieurs fascicules de voyage sous les yeux. Le critère de choix, au départ est « Je veux du soleil ». On va donc saisir le premier livre et prendre connaissance de quelques hôtels situés dans des villes ensoleillées dont on notera les noms sur un bout de papier. Mais lors de cette lecture, on tombe sur le mot piscine qui paraît assez intéressant. On oriente alors ses recherches sur les hôtels avec piscine, au soleil. Mais on doit vérifier si les hôtels qu'on a déjà sélectionnés répondent bien à cette nouvelle condition. Toujours lors de cette consultation, après avoir pris note de quelques nouveaux hôtels, ce sont les mots « à visiter dans la région » qui attire l'attention, car quelques visites pourraient agrémenter le séjour; il faut donc réorienter les recherches et vérifier les noms déjà sélectionnés. Après ces raffinements successifs, la découverte d'un autre paramètre pourrait totalement modifier la recherche: la liste des prix située à la fin des catalogues. En comparant ceux des hôtels retenus avec la bourse disponible, il faut se rendre à l'évidence: les deux ne sont pas forcément compatibles et il faudra donc peut-être tout reparcourir pour pouvoir opérer un choix valide.

Ces deux exemples ont simplement le mérite de montrer combien cette tâche de recherche d'informations peut prendre des formes radicalement différentes: soit très structurée, soit fortement non structurée ou plus exactement dépendante des étapes précédentes de la recherche. La difficulté est que même si l'utilisateur a, au départ, quelques critères de base, ceux-ci peuvent devenir marginaux au fur et à mesure que la recherche progresse, pour laisser la place à d'autres priorités.

Dans la tâche qui nous occupe, les utilisateurs peuvent rechercher deux grands types d'informations: celles relatives à Strasbourg et à l'IECS et celles concernant les universités étrangères. Dans les deux cas, la recherche d'informations désirées suit la même démarche. La première étape consiste à savoir où l'on peut trouver ces informations car il existe plusieurs points sur le campus où des informations sont disponibles, mais toutes les informations ne sont pas disponibles à tous ces points. Lorsque les différents points intéressants sont repérés, il suffit alors de s'y rendre et de poser les questions aux responsables. En général, on se retrouve rapidement avec une collection d'une demi-douzaine de petits ouvrages dans lesquels il doit être possible de trouver son bonheur. La deuxième étape de la démarche commence alors: c'est l'identification des informations. Dans le cas qui nous occupe, nous nous situons un peu entre les deux exemples donnés ci-dessus. En effet, les étudiants ont, sauf exception, une idée assez précise de ce qu'ils recherchent. La quête des informations se fera donc par une recherche par raffinements successifs. Dans un premier temps, la personne va s'orienter vers un pays, puis une ville et enfin une université. Ensuite, elle devra s'orienter vers les renseignements qu'elle désire obtenir: des informations sur les logements, les musées, les cours donnés... Plusieurs

possibilités sont alors envisageables (cette liste n'est pas exhaustive, mais elle nous semble couvrir les cas les plus probables):

1. aucune information n'est trouvée, il faut recommencer le processus depuis le début, avec d'autres éléments de recherche;
2. la recherche initiale n'aboutit pas, mais une information de substitution est trouvée (par exemple, je recherche une information sur les universités suédoises, il n'y en a pas, par contre, je tombe sur une université danoise qui me donne satisfaction);
3. la recherche initiale aboutit;
4. la recherche initiale aboutit et en plus, j'ai des informations complémentaires sur le sujet.

La durée pour la réalisation complète de cette tâche peut être estimée en moyenne entre cinq et quinze minutes.

c. Apports du monde cinématographique

Pour présenter notre scénario, nous allons nous référer, en plus de l'analyse de la tâche, à la comparaison que nous avons effectuée avec le scénario cinématographique (voir 5.1).

(1) Les éléments

Les éléments d'un scénario cinématographique sont: l'histoire et la narration, l'unité de sujet, l'intrigue secondaire et le dialogue.

Si on examine les éléments de notre scénario, on peut y voir l'information qui va être présentée à l'utilisateur (l'histoire) et la façon dont on va organiser cette présentation (la narration). Les renseignements sur Strasbourg et ses universités partenaires représentent le sujet principal du scénario. A côté de cela, le scénario possède une intrigue secondaire: à certains logos, nous avons associé du son afin de distraire l'utilisateur et de le renseigner sur l'origine de ces logos. Le dernier élément est le dialogue entre l'utilisateur et la borne. Grâce à lui, l'utilisateur va pouvoir avancer dans sa consultation en répondant par geste.

(2) Les parties

Les parties d'un scénario cinématographique sont: l'exposition, le climax, la loi de progression continue, les scènes et séquences et l'accroche.

Notre scénario comprend plusieurs parties. Le menu principal constitue l'exposition, c'est-à-dire le point de départ de la recherche. Il n'y a pas de possibilité d'apprentissage du logiciel car l'utilisation est assez simple et ne nécessite pas de grandes explications. Si on reprend la définition retenue (transposée de l'exposé de la loi de progression continue), notre scénario est une série d'écrans menant à l'information recherchée. Un écran peut être considéré comme une

scène et un ensemble d'écrans, une séquence. Au cinéma, les séquences sont fixes. Dans notre scénario, il existe des liens fixes entre scènes mais les séquences varient lors de chaque utilisation du logiciel. Le graphe de transition (voir Figure 5-1) reprend les différents écrans de Télémaque en les présentant comme les noeuds d'un graphe et indique les liens entre tous ceux-ci. Par exemple, une scène est "Menu de la ville" et une séquence pourrait être {Choix de la langue, Menu principal, Menu de la ville, Présentation de la ville}. Cette séquence représente une navigation fixe dans Télémaque. Le graphe de transition permet donc de voir la navigation fixe permise entre les différents écrans du scénario. Par contre, il ne tient pas compte de la navigation transversale qu'autorisent certaines fonctionnalités du logiciel c'est-à-dire qu'il est possible de passer d'un écran vers un autre écran qui ne lui est pas adjacent dans le graphe de transition.

Une séquence issue d'une navigation transversale pourrait être {Choix de la langue, Menu principal, Menu de la ville, Menu de l'université, Menu du département, Menu principal}. Comme nous l'avons annoncé dans l'analyse de la tâche, on peut facilement voir apparaître le concept de recherche par raffinements successifs. En effet, chaque passage d'un noeud amont vers un noeud aval restreint le champ d'informations que l'utilisateur va avoir à sa disposition. La dernière partie du scénario est l'écran de présentation qui peut être considéré comme une accroche. Cet écran est actif lorsqu'il n'y a aucun utilisateur devant la borne et c'est donc lui qui doit attirer l'attention.

(3) Les procédés de narration

Notre scénario respecte certaines règles de dramatisation. Au cinéma, ces règles de dramatisation sont: la concentration de l'histoire, l'émotionnalisation, la hiérarchisation et la création d'une courbe. Ainsi, nous avons concentré l'information pour respecter les capacités d'assimilation des utilisateurs (une session ne dure généralement pas plus de quelques minutes). Nous avons également essayé de créer des variations d'intensité afin de ne pas lasser l'utilisateur. Pour ce faire, nous avons alterné les média lors de la présentation d'informations.

d. Caractéristiques du scénario de notre borne

Nous allons montrer comment les facteurs de [LOE92] (voir 5.3.2) ont influencé notre scénario. Les utilisateurs de Télémaque peuvent être novices et suivre le parcours normal de consultation ou experts et utiliser des fonctions de raccourci pour trouver l'information désirée. La durée de vie de l'information (qui est généralement d'une année académique) n'entraîne pas de contraintes au niveau du temps de réponse. Le scénario offre une restitution dépendante des requêtes de l'utilisateur et un feed-back immédiat. Vu que les sessions d'un même utilisateur ne sont pas nombreuses et qu'il n'y a pas de mécanisme d'identification, aucun historique des préférences n'est conservé.

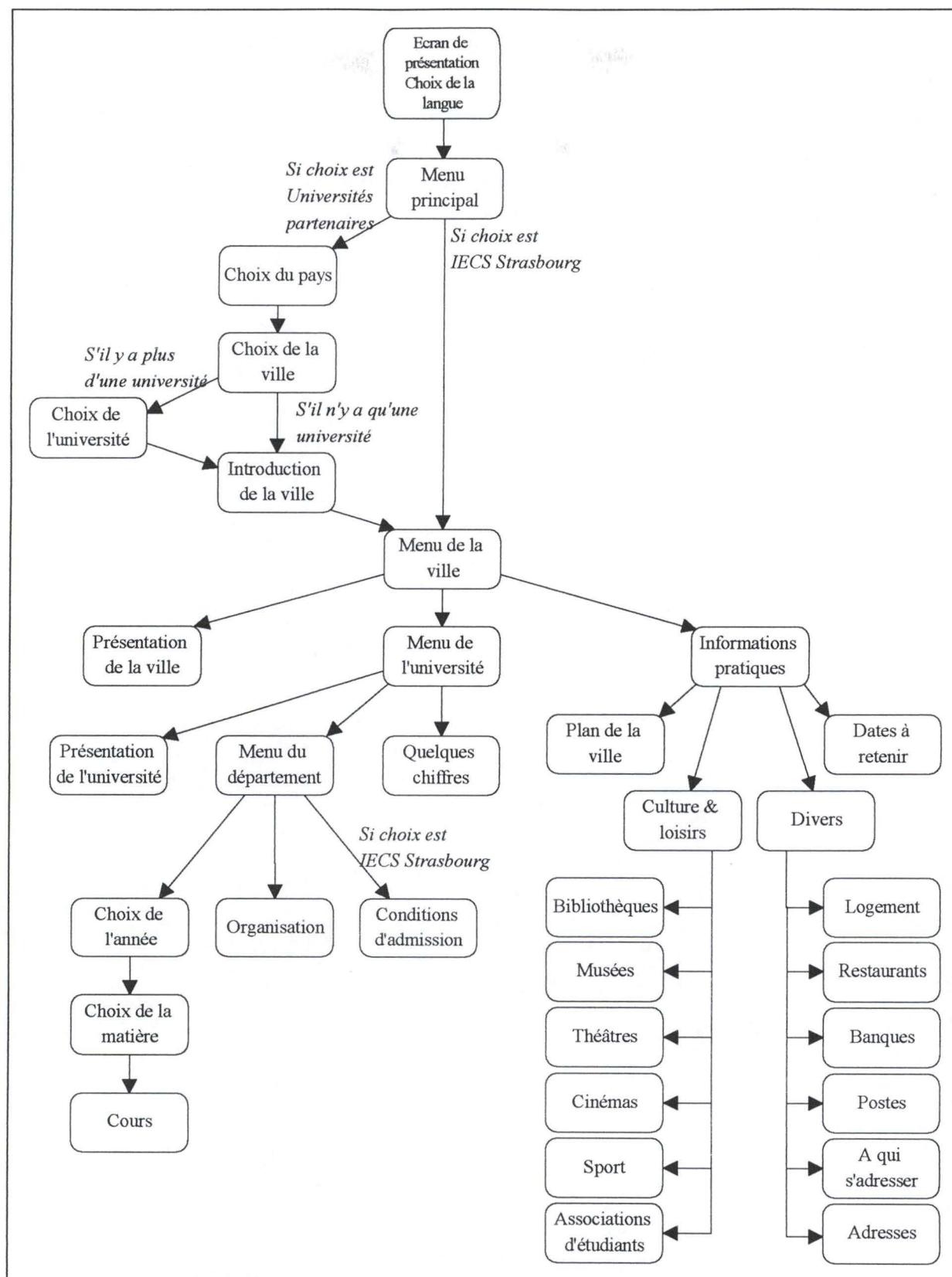


Figure 5-1. Graphe de transition de Télémaque

Après avoir présenté le scénario de notre logiciel c'est-à-dire la navigation permise entre les différents écrans, nous allons, dans le chapitre suivant, étudier d'un peu plus près la conception de ces écrans et des autres interfaces. Les interfaces devront contenir les fonctions nécessaires pour permettre le parcours établi dans le scénario ainsi que certaines possibilités de navigation transversale. De plus, elles devront respecter un certain nombre de règles propres aux logiciels classiques ou aux bornes interactives multimédia.

6. Les interfaces de Télémaque

Dans ce chapitre, nous commencerons, dans la section 6.1, par définir quelques notions qui nous permettront de caractériser et de comprendre la conception de nos écrans. Car, si les écrans d'un logiciel doivent être esthétiques, ce n'est certainement pas la seule condition qu'ils doivent remplir pour être corrects d'un point de vue ergonomique. Ensuite, dans la section 6.2, nous étudierons les écrans de Télémaque et nous les évaluerons à l'aide des critères de Shneiderman dans la section 6.3. Nous terminerons ce chapitre en expliquant, dans la section 6.4, la conception des autres types d'interfaces (le premier type étant les écrans) autrement dit les différents média présents dans Télémaque.

6.1. Rappel de quelques notions d'ergonomie

Suivant la définition donnée dans [KAR86], on peut dire que *« l'ergonomie a été définie comme l'étude des aspects anatomiques, physiologiques et psychologiques de l'homme au travail, son objectif essentiel étant l'adaptation du travail à l'homme »*. Cette discipline est née après la seconde guerre mondiale. Au départ, ses préoccupations principales étaient productivistes: *« son but premier est d'améliorer l'efficacité, la santé et le confort du travailleur ou de l'ensemble du système "homme - machine - environnement" »* [KAR86]. Mais l'ergonomie logicielle qui nous intéresse plus particulièrement est arrivée bien après cette époque. C'est au début des années 80 qu'elle est apparue. A ce moment, elle ne se préoccupait que d'un niveau assez sommaire des écrans (la forme et la disposition des caractères, la luminance des écrans, etc.). C'est seulement quelques années plus tard que des préoccupations plus fines sont apparues (création de procédures plus souples qui peuvent s'adapter à l'utilisateur, etc.).

Les différents points que nous proposons ici sont inspirés de [BOD89]. Nous traiterons d'une définition d'interface, du modèle de Norman qui nous permettra d'introduire les notions de distances sémantique et articulatoire. Pour terminer, nous présenterons le degré d'implication de l'utilisateur.

6.1.1. Qu'est-ce qu'une interface?

« Une interface est un système informatique utilisé par une personne pour réaliser une tâche accomplie à l'aide d'un ensemble de moyens informatiques » [BOD89]. La raison d'être de cette interface est donc de permettre à l'utilisateur d'accomplir avec le plus d'aisance possible cette tâche.

On peut voir que deux concepts s'opposent: la tâche et l'interface. Pour être plus précis, nous pouvons reprendre les termes de [BOD89], *« l'univers psychologique de la tâche »* et *« l'univers physique de l'interface »*.

C'est-à-dire que d'une part, l'utilisateur se représente ses objectifs dans l'univers psychologique d'une tâche à effectuer et d'autre part, l'utilisateur doit traduire ses objectifs dans des termes qui existent au niveau de l'univers physique de l'interface. Exemple: une personne pense écrire sur une feuille de papier quadrillée des nombres dans une colonne, puis en faire la somme et l'inscrire au bas de cette colonne et cette même personne a devant elle un tableur. Pour réaliser sa tâche, elle devra traduire les faits identifiés dans l'univers psychologique de la tâche, en termes physiques de l'interface (sélectionner une cellule à l'aide de la souris, taper un montant et valider avec la touche return, etc.).

Le souci principal de conception sera donc de réduire le fossé qui existe entre ces deux univers. Pour mieux voir où se situe la distance qui existe entre ces deux mondes et comment la diminuer, nous pouvons considérer les deux modèles élaborés par D.A. Norman [NOR89]: le modèle de la tâche et le modèle des deux golfes. Nous verrons en détail le premier, le second étant, sur les bases du premier, une sorte d'appareil de mesure des distances présentes.

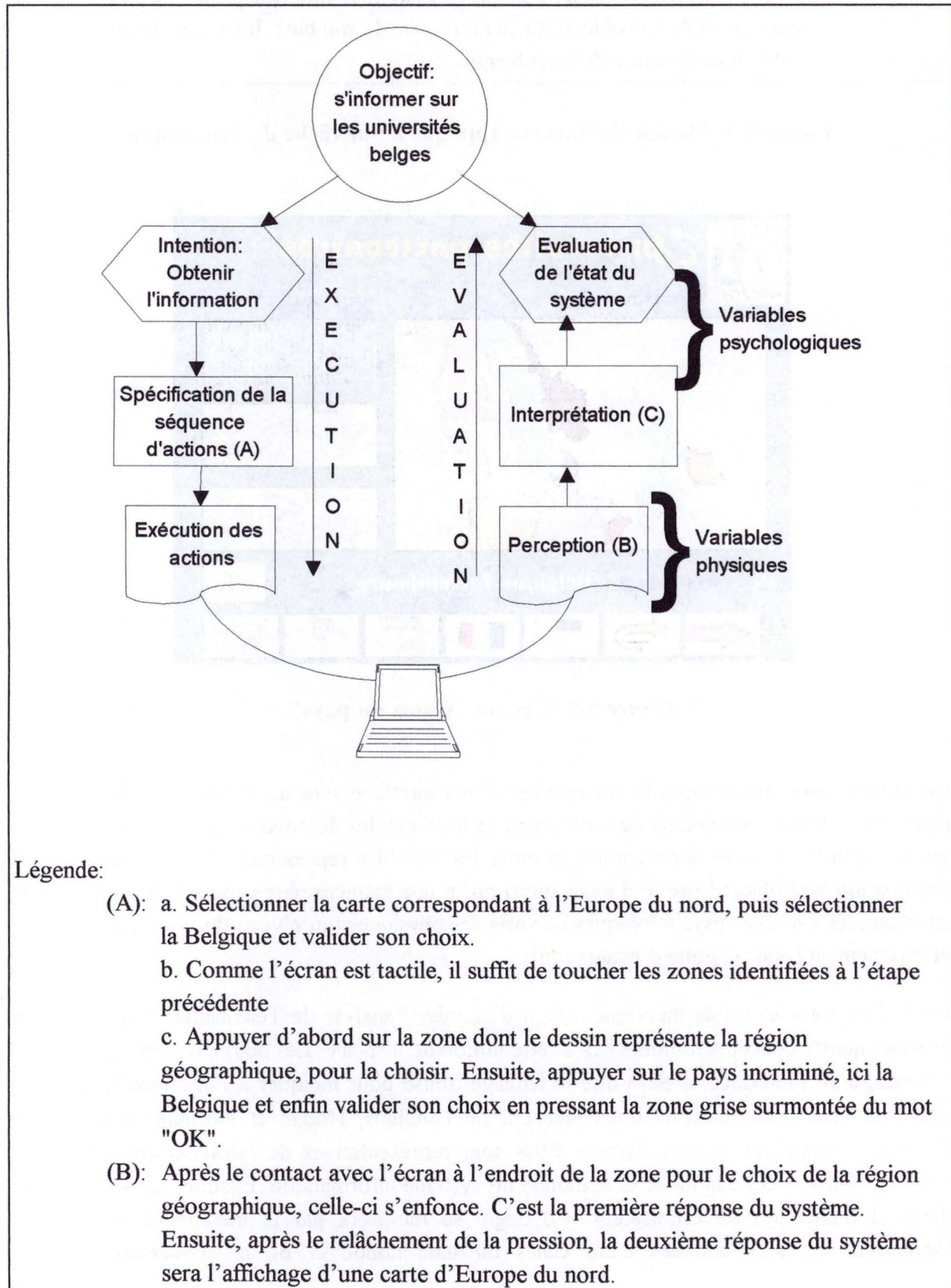
6.1.2. Modèle de Norman

Ce modèle décompose une tâche à accomplir, face à un système informatique, en sept étapes distinctes.

1. Identification des objectifs.
2. Définition de l'intention: c'est la décision d'agir pour réaliser les objectifs définis à l'étape précédente.
3. Spécification de la séquence d'actions, c'est-à-dire:
 - a. transposer ses objectifs au niveau du système;
 - b. déterminer les mécanismes de contrôle mis à la disposition pour obtenir l'état désiré;
 - c. trouver les manipulations requises de ces mécanismes.
4. Exécution de la séquence d'actions identifiée au point précédent.
5. Perception de l'état du système: c'est-à-dire qu'il faut observer comment le système réagit face à l'activation de ses mécanismes de contrôle.
6. Interprétation de l'état du système: l'utilisateur doit traduire l'état physique du système en état psychologique et il va interpréter l'état perçu en fonction de ses objectifs.
7. Evaluation de l'état du système: il faut alors comparer l'interprétation de l'état perçu du système par rapport aux objectifs initiaux pour savoir si le résultat obtenu est conforme aux attentes initiales ou s'il faut refaire le circuit.

A la Figure 6-1, vous pouvez voir ce modèle appliqué à un exemple repris dans Télémaque. Il s'agit d'atteindre les informations concernant les universités belges. Vous pouvez consulter l'écran auquel on se réfère pour cette tâche à la Figure 6-2.

A la Figure 6-1, vous pouvez voir ce modèle appliqué à un exemple repris dans Télémaque. Il s'agit d'atteindre les informations concernant les universités belges. Vous pouvez consulter l'écran auquel on se réfère pour cette tâche à la Figure 6-2.



- (C): Après avoir appuyé sur la zone pour le choix de la région géographique, une carte d'Europe du nord s'affiche, avec une phrase: « Choisissez le pays ». L'utilisateur va donc traduire l'état physique qui est l'apparition d'un élément graphique dans l'écran et d'une proposition, en une étape franchie pour la réalisation de son objectif (il se rapproche de son but). Il va alors interpréter cet état en fonction de son objectif

Figure 6-1. Modèle de Norman appliqué à une tâche de Télémaque

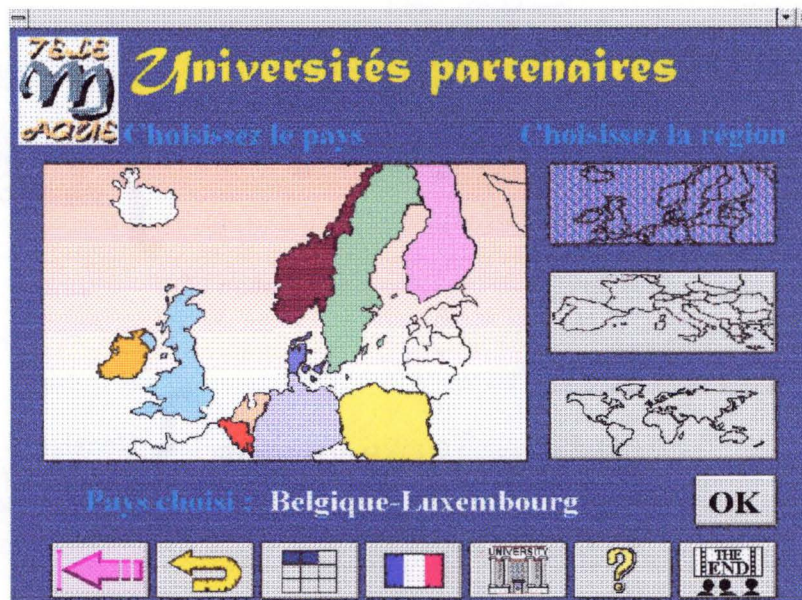


Figure 6-2. L'écran "Choix du pays"

Ce modèle nous montre que le but premier d'une interface sera de diminuer la distance qui peut exister entre les objectifs de l'utilisateur et les variables de contrôle du système (c'est ce qu'on appellera le golfe d'exécution) et entre les variables représentant l'état du système et l'évaluation de l'objectif (golfe d'évaluation) ou, d'une manière plus générale, de transformer au mieux les variables psychologiques en variables physiques (au niveau du golfe d'exécution) et inversement (pour le golfe d'évaluation).

Pour avoir tous les outils théoriques avant d'aborder l'analyse de Télémaque, il nous reste à analyser quatre concepts toujours liés à cette notion de distance. Les deux premiers concernent le langage de l'interface, c'est-à-dire le langage utilisé pour indiquer à l'interface les actions que l'on veut accomplir. Ils nous viennent de Hutchins, Hollan et Norman: il s'agit des distances sémantique et articulatoire. Elles sont représentatives de l'éloignement qui existe entre le monde réel de la tâche et le monde du système informatique. Ensuite, nous verrons le degré d'implication de l'utilisateur. Ce degré se mesurera par la métaphore utilisée par l'interface, les plus connues étant celles du mini-monde et de la conversation. Nous

expliciterons également une troisième métaphore proposée dans [VAN95]. Après les avoir présentées, nous tenterons de montrer à laquelle de ces métaphores Télémaque appartient.

6.1.3. Distances contenues dans le langage de l'interface

Par cette notion de distance, nous voulons définir l'effort cognitif réalisé par l'utilisateur pour effectuer une tâche. Celui-ci sera mesuré par l'éloignement qui existe entre le travail à accomplir "normalement" et ce même travail réalisé via l'interface d'un système informatique. Plus les deux tâches seront similaires et moins l'effort cognitif sera grand.

Selon Hutchins [BOD89], on peut distinguer la distance sémantique et la distance articulatoire (voir Figure 6-3).

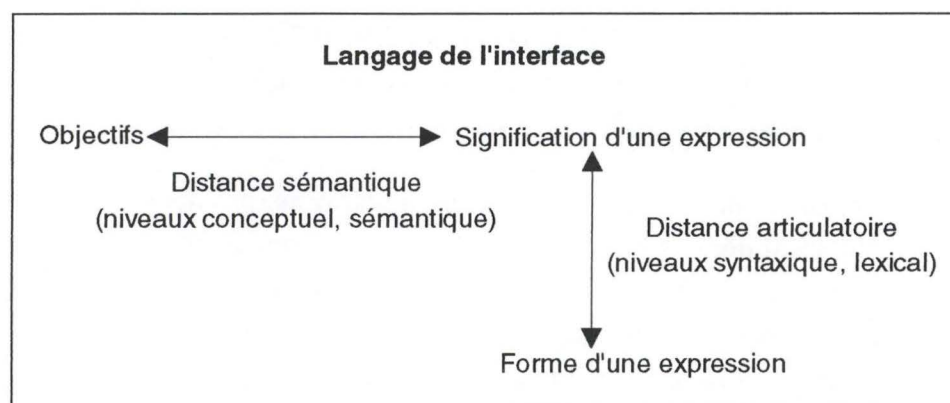


Figure 6-3. Distances sémantique et articulatoire

a. Distance sémantique

Selon [BOD89], la distance sémantique provient de choix effectués à deux niveaux d'abstraction: au niveau conceptuel et au niveau sémantique. Au premier niveau, il faut observer les objets du langage de l'interface, les relations qui existent entre eux et les opérations autorisées sur ceux-ci. Il faut alors voir si tout ceci est conforme aux mêmes éléments qui caractérisent la tâche de l'utilisateur, s'il « y a adéquation entre le modèle conceptuel que l'utilisateur a de la tâche et celui qu'il se fait de l'interface » [BOD89]. Pour le second niveau, il suffit de voir s'il faut beaucoup d'informations pour décrire une action autorisée par l'interface et si ces informations sont étrangères ou non à la tâche.

Un des buts du concepteur d'interfaces sera donc de diminuer cette distance. Pour le faire, il a quatre grandes possibilités:

1. utiliser des langages de haut niveau qui permettent de décrire la tâche au niveau de l'interface dans le même langage que celui utilisé dans le domaine de la tâche elle-même;

2. obtenir en sortie des informations qui expriment directement les concepts sémantiques;
3. adapter l'utilisateur au modèle conceptuel du système. Ce n'est donc pas le système qui s'adapte à l'utilisateur, mais bien le contraire;
4. offrir une bonne assistance à l'utilisateur.

Dans le cas d'une borne interactive informative, ces quatre points ne sont pas équivalents. En effet, par la nature même d'un tel logiciel, l'utilisation d'un langage de haut niveau sera généralement inappropriée. Et bien entendu, la solution consistant à vouloir adapter l'utilisateur au modèle conceptuel n'est pas envisageable.

b. Distance articulatoire

Elle représente la distance qui existe entre les significations des expressions du langage et leur forme physique. Et toujours d'après [BOD89], elle dépend des choix effectués ici aussi à deux niveaux d'abstraction: le niveau syntaxique et le niveau lexical.

Au niveau syntaxique, on prendra en compte le choix de la forme physique des dispositifs de contrôle (par exemple, les différents boutons ou listes de sélection) et de commande, le choix d'un style d'interaction (par exemple, le langage naturel, le langage de requêtes, les touches de fonction, les menus ou encore la manipulation directe) et la représentation des retours d'information (contrôle des entrées, de la durée, etc.).

Au niveau lexical, on s'arrêtera plutôt sur les dénominations utilisées pour les commandes, les menus ou les abréviations, sur la définition des formats (icônes, curseurs, etc.) et sur les codages utilisés (signification des couleurs, des clignotements, de la typographie, etc.).

Nous allons maintenant nous pencher sur les rapports qui peuvent exister entre un utilisateur et des interfaces. Nous en distinguerons trois types que nous définirons brièvement.

6.1.4. Degré d'implication directe de l'utilisateur

La facilité qu'éprouvera un utilisateur à travailler avec un système informatique dépend aussi du degré d'implication de cet utilisateur dans l'environnement informatisé mis à sa disposition. Dans [LAU86], deux types différents d'interfaces ont été identifiées: les interfaces basées sur la métaphore du mini-monde et celles construites sur la métaphore de la conversation. En plus de ces deux métaphores traditionnelles, on peut en tirer une troisième de [VAN95]: la métaphore navigationnelle.

a. Métaphore du mini-monde

Dans ce premier cas, nous pouvons dire que l'utilisateur devient un acteur et qu'il tient le rôle principal dans un scénario. En fait, l'interface devient le monde dans lequel l'utilisateur va pouvoir agir directement sur les objets qui le composent.

Pour que l'interface joue pleinement ce rôle, plusieurs conditions doivent être respectées:

- les distances sémantique et articulatoire doivent être très petites;
- les langages d'input et d'output doivent pouvoir utiliser les mêmes objets, pour donner l'impression à l'utilisateur que l'expression d'output est le véritable "objet", et que l'opération appliquée l'est directement sur cet objet et non sur une de ces représentations;
- le système doit donner ses réponses en temps réel, pour éliminer l'impression que le système n'est qu'un intermédiaire;
- l'interface doit être aussi discrète que possible quant à ses interventions au niveau des messages d'erreurs, de l'aide. L'apparition de messages au milieu de l'écran risque en effet de perturber l'utilisateur qui se trouvait dans un "autre monde".

b. Métaphore de la conversation

Dans le cas de l'interface construite sur la métaphore de la conversation, la relation entre l'utilisateur et l'interface est tout à fait différente. Ici, l'interface joue un rôle d'intermédiaire entre l'utilisateur et l'ordinateur. L'utilisateur conversera avec l'interface qui transmettra les ordres de celui-ci à l'ordinateur.

c. La métaphore navigationnelle

Une interface est caractérisée comme telle lorsqu'elle aide l'utilisateur à se mouvoir dans un espace d'informations multimédia. Nous pouvons considérer celle-ci comme une spécialisation de la métaphore de la conversation.

C'est bien entendu dans cette troisième optique que les interfaces de Télémaque ont été réalisées. Lors de son utilisation, Télémaque servira de support pour permettre à l'utilisateur d'atteindre les informations désirées.

6.2. Etude des écrans de Télémaque

Après ces quelques définitions, nous allons pouvoir étudier les écrans présents dans Télémaque. Lors de cette étude, nous nous proposons de détailler dans une première partie les moyens d'interaction qui nous intéressent ainsi que quelques objets interactifs abstraits. Après

cela, nous verrons les critères ergonomiques de design sur lesquels nous nous sommes basés pour réaliser nos écrans. Nous terminerons cette partie en présentant les éléments communs à la plupart des écrans, puis nous étudierons la spécificité de chacun des écrans. Nous en présenterons les objets interactifs et nous expliquerons en quoi leur utilisation diminue les différentes distances.

6.2.1. Les moyens d'interaction

Un moyen d'interaction est constitué d'un environnement matériel et logiciel à l'aide duquel l'utilisateur capture de l'information ou la restitue. Les différentes caractéristiques de ceux-ci ont été tirées de [VAN94].

Le plus connu des moyens d'interaction est bien entendu la souris. Mais dans le cas de notre borne interactive, il ne nous semblait pas très adapté. Nous devons en effet assurer la solidité du matériel et la présence d'un périphérique fragile ne nous le permettait pas. Par ailleurs, il nous est apparu que désigner un objet à l'aide du doigt plutôt qu'avec une souris diminuait la distance articulaire, puisque l'on supprimait la plupart des interférences entre le monde de la tâche et l'univers du système informatique. Ce sont principalement ces deux raisons qui nous ont poussés à choisir, comme moyen d'interaction, un écran tactile. Mais son utilisation impose quelques contraintes plus ou moins évidentes. Tout d'abord, la taille des objets interactifs doit être adaptée. De plus, il est recommandé d'offrir un feed-back direct à toutes les sélections effectuées (un bip sonore par exemple) pour signaler qu'elles ont été réalisées. Enfin, un inconvénient moins évident au premier abord: l'absence de curseur, qui empêche l'utilisation d'un curseur contextuel souvent bien utile. Pour conclure, on peut signaler que l'utilisation de l'écran tactile ne doit pas être trop longue à cause de la fatigue musculaire du bras qu'elle occasionne.

Dans notre borne, nous utilisons également comme moyens d'interaction la synthèse vocale, le son, les images statiques et les images vidéos (appelés des moyens d'interaction multimédia). A chacun d'entre eux sont associées des règles plus ou moins strictes pour assurer leur bonne utilisation. A titre d'exemple, voici quelques règles concernant les images statiques [VAN94].

- ◇ Tout affichage d'image doit tenir compte de la distance de lecture et de la lumière ambiante.
- ◇ Tout affichage doit tenir compte de la résolution, des détails et du bruit de l'image.
- ◇ Les proportions de toute image doivent être compatibles (exemple: une carte géographique ne doit ni être trop allongée, ni trop étirée).

Si des critères tels que le deuxième ou le troisième peuvent être facilement contrôlables par un choix judicieux des images et une numérisation de qualité, le premier critère peut être plus difficilement réalisé puisqu'il fait intervenir un élément de l'environnement qui peut varier (la lumière ambiante).

6.2.2. Définition des principaux objets interactifs abstraits de Télémaque

Les définitions que nous présentons ici sont tirées de [VAN92].

a. Icône

L'icône est un graphique généralement simple qui a pour unique fonction de représenter une action, une fonction ou une idée à l'utilisateur. Par exemple, un point d'interrogation va faire penser à de l'aide, une disquette fera référence à une sauvegarde...

Il n'y a donc aucune action directement associée à une icône.

b. Menu

Un menu permet à l'utilisateur de déclencher toute une série de fonctions ou d'actions du système. Il peut être qualifié de textuel (si les éléments qui le composent sont uniquement du texte), d'iconique ou de graphique (si les éléments qui le composent sont uniquement des icônes) ou de mixte.

c. Bouton de commande

Le bouton de commande est un contrôle que l'on utilise pour déclencher une action. Celle-ci est identifiée par le libellé qui se situe sur le bouton.

d. Bouton graphique

Celui-ci ressemble au précédent. La différence se situe dans le fait que le libellé peut être soit remplacé, soit renforcé par une icône.

e. Liste de sélection

Une liste de sélection est une liste d'éléments que l'utilisateur peut parcourir et où il peut faire une ou plusieurs sélections. Une liste de sélection peut également être qualifiée de textuelle, de graphique ou de mixte.

6.2.3. Critères ergonomiques de design

Avant de présenter la structure des écrans de Télémaque, nous allons présenter les critères ergonomiques de design de [VAN94] qui ont servi de base pour élaborer nos écrans. Ils sont au nombre de huit et proviennent de différents ouvrages. Ce sont des critères dits a priori. En effet, ils peuvent servir de règles de base à respecter pour construire des écrans corrects d'un point de vue ergonomique. Nous allons voir s'ils sont tous pertinents face à une borne interactive multimédia et les définir par rapport à un tel logiciel puis nous présenterons leurs implications sur Télémaque. Pour bien comprendre nos choix, il convient de ne pas perdre de vue les caractéristiques d'un écran tactile et de consulter, s'il le faut, l'analyse de la tâche

effectuée à la section 5.3.3. Dans une troisième section, nous donnerons quelques points à ne pas négliger lors du développement d'un logiciel multimédia utilisant un écran tactile.

a. Définitions et pertinence

Lors de ces définitions, nous décomposerons les différents critères en niveaux linguistiques. Ces décompositions seront revues par rapport à leur origine dans [VAN94], pour essayer de mieux répondre aux intérêts des bornes interactives.

Les huit critères de conception que nous avons retenus sont:

(1) La compatibilité

Ce critère concerne la facilité avec laquelle a lieu le passage d'informations et d'actions du monde réel vers le système informatique au travers de l'interface homme-machine. Il existe plusieurs types de compatibilité. Nous retiendrons:

- ◇ la **compatibilité syntaxique opérationnelle**: elle concerne l'ordre des opérations, les procédures que l'utilisateur a l'habitude de faire pour réaliser sa tâche;
- ◇ la **compatibilité lexicale**, composée de la **compatibilité grammaticale** (utilisation de la terminologie employée par l'utilisateur), la **compatibilité linguistique** (compatibilité avec la langue naturelle de l'utilisateur) et la **compatibilité de standard** (par exemple, utilisation d'objets qui répondent à des normes reconnues comme des standards).

(2) La cohérence

Une interface homme-machine peut être caractérisée de cohérente si les données et les actions peuvent facilement être identifiées et utilisées, c'est-à-dire si, pour réaliser une tâche, un utilisateur peut facilement percevoir ce qu'il doit faire. Ce critère peut se décomposer pour nous en:

- ◇ **cohérence sémantique**, c'est la cohérence du sens attribué aux objets, à leurs propriétés et relations (par exemple, au sein d'un logiciel, une même couleur est toujours utilisée pour afficher des informations d'un même type);
- ◇ **cohérence lexicale** qui concerne la cohérence du choix et de l'utilisation des libellés, des dénominations des items de menus... Elle se compose de la **cohérence spatiale** (la cohérence dans les placements des objets interactifs concrets) et la **cohérence grammaticale** (la cohérence dans la structure des libellés et des différentes dénominations);
- ◇ **cohérence physique**, c'est la cohérence des éléments tels que la lumière (les couleurs, le clignotement, etc.) ou le son (le volume, la fréquence, etc.).

(3) La charge de travail

La charge de travail concerne la quantité de données à utiliser pour réaliser une tâche. Si elle est réduite, l'interface homme-machine sera considérée comme bonne. Ce critère prend en compte:

- ◇ la **brièveté** qui se décompose en **concision** (charge informationnelle véhiculée par les saisies et les affichages de données) et en **actions minimales** (charge informationnelle véhiculée par le nombre minimal d'actions à effectuer pour accomplir une tâche);
- ◇ la **charge mentale**: c'est une notion empirique qui ne possède pas de définition. Pour la comprendre, nous pouvons signaler comment la réduire. Il faut respecter la capacité de la mémoire à court terme, les facultés de vision et d'audition de l'utilisateur;
- ◇ la **charge symbolique**: ici aussi, cette notion est empirique. Pour obtenir une bonne charge symbolique, il faut limiter le nombre de symboles utilisés;
- ◇ le **respect perceptuel**: c'est-à-dire la prise en compte des facultés de vision et d'ouïe de l'être humain.

Ce critère est très important dans le monde des bornes interactives. Il est en effet indispensable d'alléger au minimum la charge de travail car l'utilisateur est généralement novice tant dans le sujet traité que dans l'utilisation d'une borne interactive. Il est donc intéressant de minimiser la charge de travail pour rendre sa tâche aisée, sans trop d'effort cognitif à effectuer.

(4) L'adaptabilité

L'adaptabilité est la capacité pour un système de se mettre au niveau de l'utilisateur pour faciliter la réalisation de la tâche entreprise. Mais, pour une borne interactive dont la durée d'utilisation est assez brève (entre trois et cinq minutes par personne), il n'est pas aisé de prendre en compte ce critère. En effet, il est difficilement envisageable de poser à l'utilisateur quelques questions pour établir son "portrait" afin qu'ensuite le système puisse lancer des écrans adéquats. Ce critère ne nous semble donc pas très adéquat dans le cas des bornes interactives, et nous ne l'appliquerons donc pas à la réalisation des écrans de Télémaque.

(5) Le contrôle de dialogue

Une interface est dite à contrôle explicite si et seulement si elle fournit l'apparence à l'utilisateur d'être placée sous son contrôle. Le but est de laisser l'utilisateur contrôler le déroulement du dialogue au maximum. Il pourra ainsi avoir l'impression de naviguer librement dans le logiciel. Le contrôle de dialogue se mesure au niveau de la **conversation** (soit par des **actions explicites**: les actions sont déclenchées par l'utilisateur; soit par des **actions implicites**: les actions sont déclenchées par le système; soit par des **actions mixtes**) et au niveau de la **présentation** (la **présentation manuelle**: l'utilisateur gère lui-même les différents affichages, leurs positions, etc.; la **présentation automatique**: le système ajuste

automatiquement la présentation sans intervention de l'utilisateur; la **présentation assistée**: le système ajuste la présentation à la demande de l'utilisateur).

(6) La représentativité

Une interface est qualifiée de représentative si les items de menu, les libellés et les icônes facilitent la rétention de leur sémantique. Nous retiendrons ici la **représentativité dans la conversation** (les **raccourcis**: les icônes doivent être facilement associables à la fonction ou l'idée qui leur est liée).

(7) Le guidage

Une interface homme-machine est efficace en guidage si elle informe en continu l'utilisateur sur le résultat de ses actions et sur sa position dans la réalisation de sa tâche. On considérera le **guidage dans la conversation**

- ◇ par **l'invitation**: informer l'utilisateur sur l'état courant et sur l'état qu'il veut atteindre;
- ◇ par le **feed-back immédiat**: donner une réponse à l'utilisateur pour une action entreprise.

(8) La gestion des erreurs

Si une interface homme-machine est robuste face aux erreurs des utilisateurs et se montre conviviale dans le traitement de celles-ci, alors elle sera qualifiée d'efficace dans le traitement de la gestion des erreurs. Cette gestion des erreurs se décompose en quatre points:

- ◇ la **protection vis à vis des erreurs**: l'interface doit être construite de telle sorte qu'elle réduise l'introduction d'erreurs;
- ◇ **l'identification des erreurs**: si une erreur se produit, le système doit essayer d'identifier la cause et l'endroit où elle s'est produite;
- ◇ **l'explicitation des erreurs**: l'interface doit être conçue de telle façon qu'elle explique l'erreur qui s'est produite et le moyen de la corriger;
- ◇ la **correction des erreurs**: la conception de l'interface doit prévoir le moyen de corriger toute erreur survenue.

Ce critère est bien entendu important dans tous les types de logiciels. Mais dans le cas des bornes interactives, il faut en relativiser la portée. En effet, les erreurs proviennent souvent de manipulations de données difficiles et longues (par exemple, des saisies). Or, dans les bornes interactives, ces opérations sont souvent réduites, voir inexistantes. Les seules erreurs qui restent donc à gérer sont liées à une mauvaise compréhension par l'utilisateur des fonctions associées aux items de menus ou aux boutons de commande. Pour les éviter, une modification

des libellés pour les rendre plus explicites semble appropriée. Par contre pour aider l'utilisateur "victime" d'un tel problème, il n'existe pas de solution vraiment satisfaisante. En effet, puisqu'il est impossible de connaître a priori la volonté de celui-ci, il est également impossible de déceler automatiquement l'erreur. La solution que nous avons retenue sera de lui permettre de revenir aisément sur le lieu de cette erreur pour qu'il puisse lui-même la corriger.

b. Télémaque et les critères de design

Maintenant que nous avons défini les huit critères de conception, nous allons montrer comment nous avons tenté de les respecter.

(1) *La compatibilité*

Dans le cas de la tâche qui nous occupe (la recherche et la consultation d'informations), la **compatibilité opérationnelle** ne sera pas très bonne puisque les opérations à effectuer dans le monde réel de la tâche et au niveau du système informatique ne sont pas identiques. Par contre, l'utilisation d'un vocabulaire courant procurera une bonne **compatibilité grammaticale** et la **compatibilité linguistique** sera elle aussi bien respectée grâce au multilinguisme de Télémaque. Nous pensons aussi que la **compatibilité de standard** sera conforme. En effet, la position des objets suivra une découpe d'écran, que nous verrons plus tard, reconnue par certains auteurs tels que [RUB86]. De plus, même si les icônes sont originales, les boutons qui leur servent de support ont la forme de ceux communément utilisés dans les logiciels sous Windows.

(2) *La cohérence*

Tous les objets interactifs concrets que l'on identifiera dans Télémaque posséderont toujours la même sémantique (**cohérence sémantique**). Leur positionnement constant indiquera une **cohérence spatiale** et la structure identique des libellés impliquera une **cohérence grammaticale**. L'utilisation que nous ferons des couleurs impliquera une certaine **cohérence physique**. Nous essayerons toujours d'associer à une même sémantique, la même couleur (par exemple, la couleur jaune est toujours utilisée pour afficher des informations de guidage pour l'utilisateur).

(3) *La charge de travail*

La mémoire à court terme de l'utilisateur sera peu mise à contribution puisque nous essaierons de replacer, dans chaque écran, un élément qui situera l'utilisateur dans son parcours. Celui-ci pourra être un logo contextuel ou encore un libellé explicatif. L'utilisateur ne devra donc pas se souvenir de trop d'éléments pour poursuivre sa recherche (**charge mentale**). En ce qui concerne la **charge symbolique**, elle peut être qualifiée de correcte, Télémaque ne contenant qu'une douzaine d'icônes. Pour que l'utilisateur puisse obtenir une information désirée, il devra parcourir environ sept écrans, ce qui représente une interaction avec, en moyenne, une dizaine d'objets interactifs concrets dont la compréhension et l'utilisation est simple (**brièveté**). Pour terminer, nous pouvons dire que le **respect perceptuel** sera validé: la taille des graphiques,

photos et libellés ainsi que le volume des sons seront, de l'avis de quelques utilisateurs, assez grands et forts pour respecter les capacités d'un utilisateur.

(4) Le contrôle du dialogue

Télémaque sera composé d'interfaces à **contrôle explicite**, c'est-à-dire que l'utilisateur aura l'impression de contrôler le dialogue avec le système. Cette attitude de contrôle est importante pour que l'utilisateur se sente en confiance face à la machine. La **présentation** sera **automatique**: nous ne voulons pas que l'utilisateur doive se concentrer sur la taille ou le positionnement de différents objets. Un autre contrôle de la présentation aurait alourdi la charge de travail, ce que nous ne voulons pas.

(5) La représentativité

Nous pensons que la **représentativité dans la conversation** dans Télémaque sera correcte puisque les icônes seront en général assez expressives (par exemple, nous utiliserons un point d'interrogation pour symboliser le système d'aide).

(6) Le guidage

Télémaque procurera des repères permanents aux utilisateurs pour que ceux-ci puissent se situer dans le scénario de la borne (**guidage dans la conversation par l'invitation**). De plus, la réaction immédiate aux actions de l'utilisateur procurera un bon feed-back à celui-ci (**guidage dans la conversation par feed-back immédiat**). Ces éléments donneront à Télémaque le critère de **guidage au niveau de la conversation**.

(7) La gestion des erreurs

Les moyens mis en oeuvre pour que l'utilisateur puisse revenir sur les lieux de son erreur seront au nombre de deux: la fonction de retour à l'écran précédent et la fonction d'historique.

Puisque Télémaque devra être placé dans un lieu public, sa robustesse aux erreurs de l'utilisateur, qui pourraient entraîner un arrêt du système, a été fortement étudiée et testée. Par contre, pour les erreurs d'utilisation plus courantes, la prise en main par le système ne se fera pas. Nous estimons qu'aucune erreur ne pourrait déstabiliser assez l'utilisateur que pour devoir intervenir de manière explicite. Mais l'utilisateur ne sera quand même pas abandonné en cas de problème. Il pourra toujours utiliser la fonction d'historique pour se repositionner sur l'écran où il a commis son erreur ou bien utiliser la fonction de retour en arrière.

c. Quelques autres repères ergonomiques

Nous ferons ici l'énumération de plusieurs règles complémentaires qu'il nous a semblé utile de suivre pour réaliser les écrans de notre borne interactive multimédia (avec écran tactile).

1. Les objets interactifs doivent être dimensionnés en conséquence. Une dimension minimale, que nous avons obtenue via quelques tests, pour un bouton de commande semble être 2.5 centimètres par 1.5 centimètres.
2. Les objets interactifs souvent utilisés doivent être positionnés en bas ou à droite de l'écran pour minimiser les mouvements du bras de l'utilisateur et éviter de cette manière que son bras ne cache une partie de l'écran. [OLI93]
3. Lorsque les objets interactifs n'ont pas une taille suffisante, il faut laisser la possibilité à l'utilisateur de pouvoir confirmer l'action entreprise par une action supplémentaire (une boîte de dialogue par exemple).
4. Le système doit offrir un feed-back après chaque action de l'utilisateur. Celui-ci doit être immédiat et peut prendre plusieurs formes: un bip sonore, un bouton qui s'enfonce, un voyant vert qui s'allume...
5. Le temps de réponse doit bien entendu être le plus court possible. Si celui-ci est de plusieurs secondes, il faut envisager la présence d'un contrôle de progression. [OLI93]
6. Après chaque phase de dialogue, le système devrait s'assurer de la présence de l'utilisateur. Si celui-ci ne répond pas à deux appels, le système devra revenir à une phase d'accroche. [OLI93]
7. Il est recommandé d'avertir l'utilisateur de l'utilisation d'un médium. Exemple: lors de l'utilisation d'une aide vocale, il faut prévenir l'utilisateur que l'aide va lui parvenir via une voix et non via du texte, comme il pourrait s'y attendre.
8. Si la tâche dépend fortement du son ou de la vidéo, alors le contrôle de ces média doit être laissé à l'utilisateur.

Après l'explication des critères qui nous ont servi de références, nous allons aborder la composition des écrans de notre logiciel. Nous présenterons d'abord les différents types d'écrans et expliquerons ensuite les caractéristiques communes de tous les écrans de Télémaque.

6.2.4. Présentation des différents types d'écrans

Nous avons regroupé les écrans de Télémaque selon leurs fonctions: les écrans de choix, les écrans avec informations terminales et les écrans de transition.

a. Les écrans de choix

Nous avons regroupé, dans cette catégorie, les écrans dans lesquels l'utilisateur doit opérer un choix. Celui-ci se fera soit entre différents boutons, dans une liste de sélection ou encore sur une carte.

b. Les écrans avec informations terminales

Ce sont les écrans contenant les informations recherchées par l'utilisateur. Lorsque celui-ci est positionné sur ce type d'écran, il ne sait pas aller plus loin et est donc obligé de revenir en arrière ou d'arrêter sa consultation.

c. Les écrans de transition

Les écrans de transition sont les écrans dans lesquels aucun choix ne doit être effectué et qui contiennent des informations non terminales. Dans ce type d'écran, l'utilisateur doit appuyer sur un bouton unique pour continuer son parcours.

6.2.5. Les caractéristiques communes des écrans de Télémaque

Nous allons maintenant voir les différentes caractéristiques que l'on pourra retrouver dans tous les écrans de Télémaque. Nous reprendrons d'abord l'utilisation des couleurs. Ensuite, nous étudierons la découpe type d'un écran de Télémaque et le contenu des différentes zones identifiées dans cette découpe.

a. L'utilisation des couleurs

Le choix des couleurs utilisées dans l'élaboration des écrans est très important en ergonomie. Pour trouver les combinaisons qu'on a utilisées, on s'est appuyé sur le Tableau 6-1, tiré de [VAN94], qui fait la correspondance entre des couleurs d'arrière-plan et les meilleures couleurs pour des textes ou objets à traits fins. Les pourcentages correspondent au nombre de personnes qui peuvent voir les traits fins ou le texte sur l'arrière-plan sans difficulté et qui n'éprouvent aucune gêne visuelle pour le faire.

Arrière-plan	Trait fin et texte
Blanc	bleu (94%), noir (63%), rouge (25%)
Noir	blanc (75%), jaune (63%)
Rouge	jaune (75%), blanc (56%), noir (44%)
Vert	noir (100%), bleu (56%), rouge (25%)
Bleu	blanc (81%), jaune (50%), cyan (25%)
Cyan	bleu (69%), noir (56%), rouge (37%)
Magenta	noir (63%), blanc (56%), rouge (37%)
Jaune	rouge (63%), bleu (63%), noir (56%)

Tableau 6-1. L'association des couleurs

Si on ajoute à ce tableau le fait que la couleur bleu repose l'oeil humain, on dispose de tous les éléments pour comprendre nos choix du point de vue des couleurs utilisées: un fond uniforme bleu et les textes, par ordre de fonction, jaune, blanc ou cyan. Les boutons sont gris clair et leurs libellés sont noirs (voir Figure 6-4).

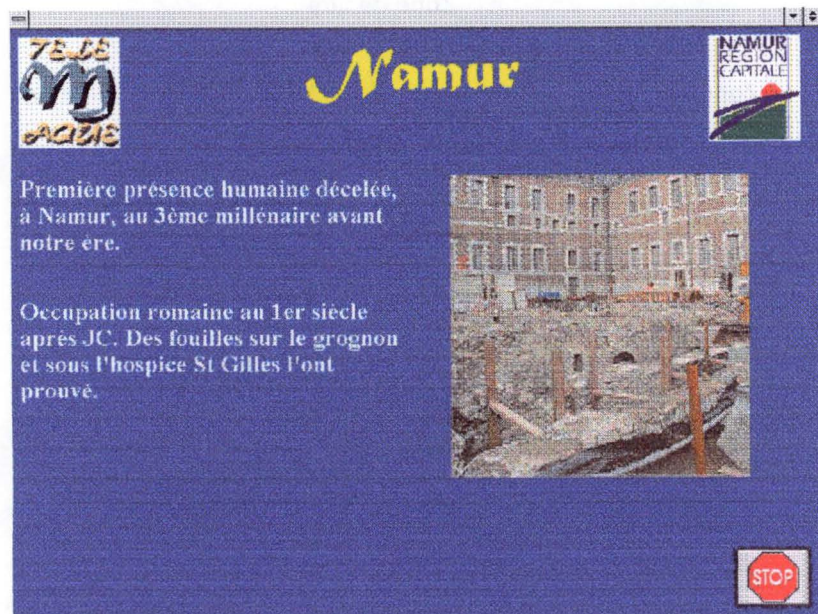


Figure 6-4. L'écran "Présentation de la ville"

Il faut également remarquer qu'une couleur est généralement associée à un endroit ou à une fonction. Par exemple, les libellés jaunes se situeront toujours en haut de l'écran et ils auront une fonction de guidage pour l'utilisateur.

b. La découpe des écrans

Tous les écrans de Télémaque (sauf l'écran de présentation) respectent un certain format décrit à la Figure 6-5.

Avant de détailler les objets qui font partie de ces différentes zones, on fera part de deux hypothèses explicatives de notre démarche.

- Chaque fois que c'était possible, nous avons utilisé des boutons graphiques. Plusieurs motivations nous ont poussés à ce choix. La première est que leur utilisation réduit la distance articulatoire. Ensuite, le multilinguisme nécessaire de notre borne impose une traduction de tous les libellés, cette traduction est naturellement inutile pour des graphiques bien choisis.
- La taille des objets présents dans les écrans est assez grande. L'utilisateur se servant de son doigt pour sélectionner des objets, il était indispensable de redimensionner ceux-ci. Leur taille a été évaluée par essais et erreurs. De plus, pour ne pas fatiguer l'utilisateur, des

libellés assez grands devaient être utilisés. Ceci renforçait donc le besoin d'utiliser des objets interactifs assez grands.

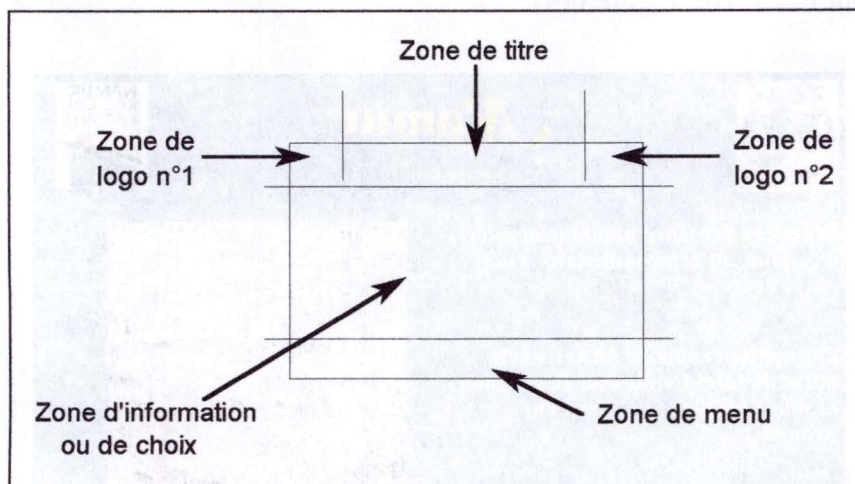


Figure 6-5. Découpe des écrans de Télémaque

Ces deux hypothèses étant faites, détaillons les différentes zones identifiées à la Figure 6-5.

1. La zone de titre. Nous y trouvons un ou deux libellés qui permettront à l'utilisateur de se situer lors de son parcours dans Télémaque: c'est une zone de guidage.

Exemple: lors de la présentation des cours de première année à l'université de Namur, les valeurs des deux libellés seront respectivement "Namur" et "Première année" (voir Figure 6-6).



Figure 6-6. La zone de titre de l'écran "Choix de la matière"

2. La zone de logo n°1. Cette zone reprend le logo de la borne Télémaque. Sa présence est généralement là pour "équilibrer" la composition des écrans. Nous avons en effet pu constater, lors de nos quelques tests utilisateurs, que les écrans "symétriques" étaient bien perçus par les utilisateurs.
3. La zone de logo n°2. Elle comprendra souvent un logo contextuel. Celui-ci sera fonction de la position de l'utilisateur dans Télémaque. A titre d'exemple, il pourra prendre la forme du drapeau d'un pays, du logo d'une ville ou d'une université...

4. La zone de menu. Celle-ci a une grande importance. En effet, si on excepte les écrans concernant la présentation des villes et des universités, elle contient le menu graphique qui est le tableau de bord de notre borne. Toutes les icônes que l'on peut voir sur les différents boutons ont été créées pour Télémaque, dans le souci d'être le plus expressif possible. En voici le détail.

Le retour au menu principal (Figure 6-7)

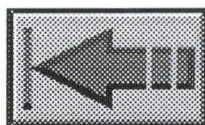


Figure 6-7. Le bouton graphique "Retour au menu principal"

Ce bouton graphique ramène l'utilisateur au niveau du menu principal (c'est-à-dire le premier menu après l'écran de présentation qui permet le choix de la langue).

Le retour à l'écran précédent (Figure 6-8)

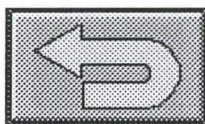


Figure 6-8. Le bouton graphique "Retour à l'écran précédent"

Ce bouton graphique permet de revenir à l'écran précédant l'écran courant.

L'historique (Figure 6-9)

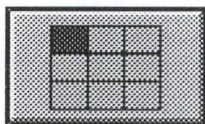


Figure 6-9. Le bouton graphique "Historique"

Ce bouton graphique est associé à une fonction d'historique. Celle-ci permet à l'utilisateur de consulter les neuf derniers écrans qu'il a vus, de la sorte que l'utilisateur peut revenir facilement sur une information déjà consultée.

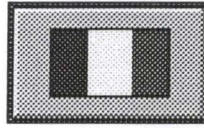
Le changement de langue (Figure 6-10)

Figure 6-10. Le bouton graphique "Changement de langue"

Ce bouton permet de visualiser la langue dans laquelle la consultation se fait actuellement, et de changer la langue utilisée. Cinq langues sont proposées dans Télémaque: le français, l'anglais, l'allemand, l'italien et l'espagnol.

Le changement d'université (Figure 6-11)

Figure 6-11. Le bouton graphique "Changement d'université"

Ce bouton permet de changer l'université à propos de laquelle l'utilisateur veut des informations. Il évite ainsi de refaire toute l'arborescence des menus. Si nous offrons cette fonction au niveau des universités, c'est parce qu'elles représentent un bon niveau d'information pour l'utilisateur (l'utilisateur veut normalement obtenir des renseignements en rapport avec une université).

L'aide (Figure 6-12)

Figure 6-12. Le bouton graphique "aide"

En appuyant sur ce bouton, une aide vocale renseigne l'utilisateur sur le fonctionnement du menu graphique. Elle lui dira : « *Appuyez sur les icônes pour connaître leurs fonctions respectives. Pour reprendre votre consultation, appuyez de nouveau sur le point d'interrogation* ». A ce moment, l'utilisateur pourra obtenir de l'aide sur le menu

graphique. Lorsqu'il voudra reprendre sa consultation, il lui suffira donc de réappuyer sur le bouton d'aide. Ce bouton est un bouton radio: cela signifie que, lorsque l'utilisateur a appuyé une première fois dessus, il se modifie. En effet, le point d'interrogation devient rouge pendant toute la durée de l'utilisation de l'aide et c'est seulement quand l'utilisateur appuiera de nouveau sur ce bouton qu'il reprendra sa couleur initiale). L'aide fournie vocalement est disponible dans les cinq langues. L'utilisation de la voix est intéressante car elle ne perturbe pas l'écran. Elle apporte de l'information sans surcharger l'écran. Elle améliore donc véritablement le logiciel.

Le bouton fin (Figure 6-13)

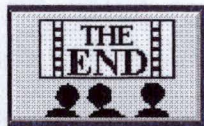


Figure 6-13. Le bouton graphique "fin"

Ce bouton termine une consultation et ramène l'utilisateur à l'écran de présentation de la borne Télémaque.

5. La zone d'information ou de choix. L'oeil est attiré par cette partie centrale de l'écran. Nous y trouverons les informations principales et les différents objets interactifs que nous détaillons dans le point suivant. C'est qu'en effet, le contenu de cette zone est spécifique à chaque type d'écran.

6.2.6. Analyse des caractéristiques spécifiques par type d'écran

On se bornera à présenter ici les objets que l'on peut trouver dans la zone d'information ou de choix des types d'écrans considérés.

a. Les écrans de choix

Les écrans de choix sont les écrans dans lesquels l'utilisateur doit opérer un choix. Celui-ci se fera soit entre différents boutons, dans une liste de sélection ou encore sur une carte. Nous présentons, dans les pages qui suivent, les différents types d'écrans de choix.

Le premier est l'écran de présentation (voir Figure 6-14). C'est en fait le premier contact que l'utilisateur va avoir avec la borne. Cet écran reprend le nom de Télémaque et trois logos en mouvement. Ces trois objets sont en fait des boutons graphiques dont l'activation déclenche une séquence sonore qui explique le logo. Un menu graphique complète cet écran. Il se compose de cinq boutons de commande à l'effigie des drapeaux de nationalité française, anglaise, allemande, italienne et espagnole et d'un libellé dynamique (il se déplace au dessus de chaque bouton de commande en étant traduit). La présence de ce libellé aidera l'utilisateur novice à agir en réduisant la distance sémantique. L'activation d'un de ces boutons emmènera l'utilisateur à l'écran nommé "Menu principal" dans la langue de son choix.



Figure 6-14. L'écran de présentation

Le deuxième écran de ce type que nous allons décrire se trouve à la Figure 6-15. Il s'agit en fait de l'écran appelé "Menu principal". Celui-ci est le premier écran où apparaissent des boutons de commande. Il en comprend deux avec les libellés "IECS Strasbourg" et "Universités partenaires". L'originalité de cet écran vient en fait des deux objets graphiques situés devant ces deux boutons de commandes: des flèches. Ici encore, la justification de la présence de ces objets est le besoin de guidance de certains utilisateurs (qui entraîne une réduction de la distance sémantique). Nous avons en effet constaté lors de nos tests que devant cet écran, quelques utilisateurs ne savaient pas comment réagir lorsque les flèches étaient absentes. Il n'était pas évident pour eux qu'il fallait appuyer sur ces "rectangles" pour poursuivre leur recherche. C'est donc forts de cette expérience que nous avons placé ces flèches qui incitent l'utilisateur à se diriger vers un de ces deux boutons.

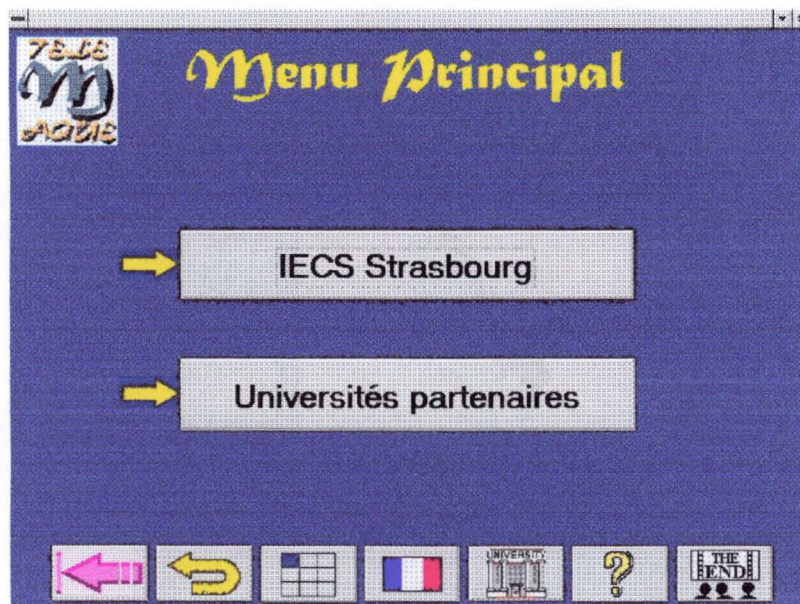


Figure 6-15. L'écran "Menu principal"

Le troisième écran que nous allons décrire est celui que l'utilisateur voit lorsqu'il doit choisir le pays avant d'orienter ses recherches vers une université. Il est représenté à la Figure 6-16. Tel que présenté, cet écran est complet. En fait, lorsqu'il apparaît réellement, tous les objets ne sont pas présents. Dans un premier temps, seuls les boutons de commandes graphiques situés sur la droite de l'écran, surmontés du libellé « Choisissez la région » sont présents. Chacun d'entre eux représente une partie du monde (Europe du nord, Europe du sud et le reste du monde). Après que l'utilisateur ait sélectionné une des régions, la carte correspondant à cette région apparaît avec la proposition « Choisissez le pays ». A ce moment, l'utilisateur devra appuyer directement sur le pays de son choix (les pays qu'il peut sélectionner, c'est-à-dire ceux où l'IECS possède au moins une université partenaire, sont coloriés). Un retour d'information direct se fera alors puisque le nom du pays sélectionné s'affichera en face des mots « Pays choisi », en même temps que l'apparition du bouton de commande "OK" pour permettre à l'utilisateur de valider son choix. Cet écran représente bien notre volonté de diminuer les distances articulatoire et sémantique. Articulatoire car le nombre d'actions à effectuer est petit

(trois) et sémantique car généralement, la représentation mentale des utilisateurs par rapport à des pays est une représentation géographique. L'utilisateur sera donc plus à son aise devant une carte que devant une liste de sélection.

Si nous avons décidé de diviser la carte du monde en trois parties, c'est essentiellement pour préserver des zones de sélection assez grandes sur la carte. Et dans cet esprit, nous avons même fusionné certains pays (par exemple la Belgique et le Luxembourg) pour obtenir une taille raisonnable.

Mais cette taille peut tout de même poser quelques problèmes surtout si les zones de sélection se juxtaposent. C'est pour cette raison que nous avons placé un bouton de validation "OK". Celui-ci permet de diminuer le nombre d'erreurs effectuées par l'utilisateur. En contrepartie, il ralentit un peu sa progression.

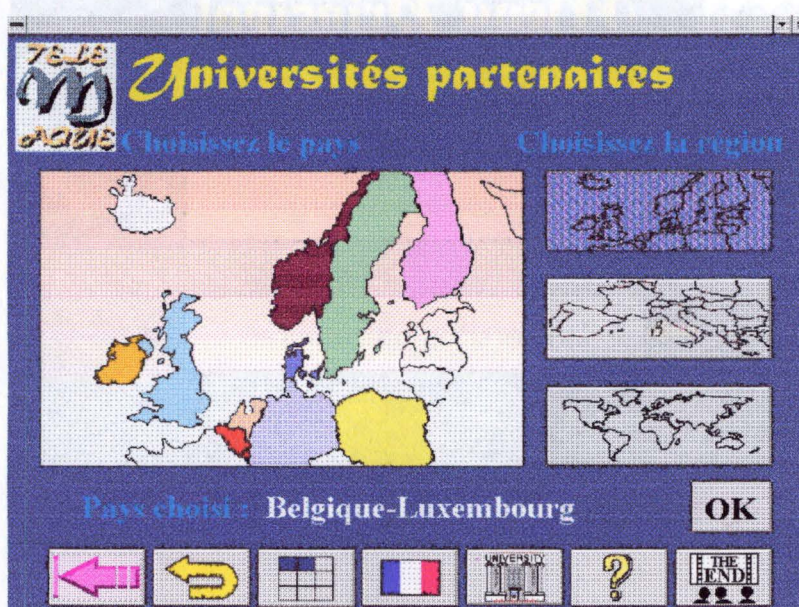


Figure 6-16. L'écran "Choix du pays"

Le quatrième écran que nous allons décrire dans la catégorie des écrans de choix est représentée à la Figure 6-17. C'est ce type d'écran que voit apparaître l'utilisateur lorsqu'il veut consulter le cursus d'une université. Comme on peut le constater, les cours sont en fait regroupés sous des intitulés collectifs (par exemple: mathématiques, moyens informatiques, etc.). Pour matérialiser ces groupes, nous avons utilisé des boutons de commandes "originaux" par leur position et leur couleur. La position des boutons de commande est dynamique, c'est-à-dire qu'elle change en fonction du nombre de boutons et ceci dans un esprit d'équilibre esthétique. La Figure 6-18 reprend les différentes positions en fonction du nombre de boutons de commande.

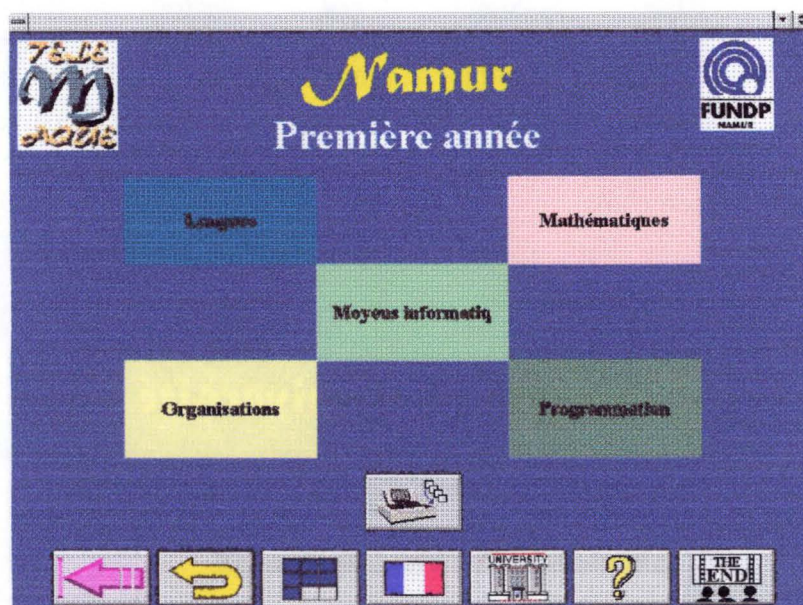


Figure 6-17. L'écran "Choix de la matière"

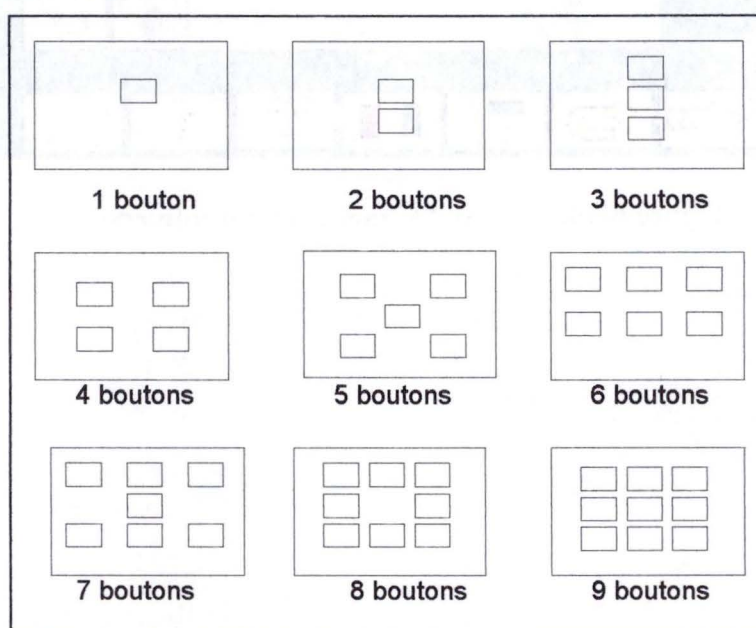


Figure 6-18. Positionnement des boutons de commande

En ce qui concerne la couleur, aucune signification sémantique n'y est attachée. Mais dans le cas d'un écran chargé (neuf boutons de commande), il nous semblait important de pouvoir facilement distinguer les neuf zones d'interaction, ce que la couleur nous a permis de faire. De plus, ces couleurs peuvent rompre une certaine monotonie présente pendant l'utilisation de la borne.

Le cinquième écran (voir Figure 6-19) vient lors de l'activation de la fonction "changement d'université" du menu graphique. Il se compose principalement d'une liste de sélection. Celle-ci reprend toutes les universités partenaires de l'IECS classées par pays. Lorsque l'utilisateur en sélectionne une, un bouton "OK" apparaît à droite de la liste de sélection pour permettre à l'utilisateur de confirmer son choix.



Figure 6-19. L'écran "Changement d'université"

Le dernier écran présenté apparaît lorsque le bouton graphique historique de la barre de menu graphique est actionné. On peut voir cet écran à la Figure 6-20. Il reprend en fait au maximum les neuf derniers écrans que l'utilisateur a consultés. Ils sont représentés par des boutons graphiques qui reprennent exactement les composants des écrans parcourus. L'utilisateur, pour retourner à un de ceux-ci, doit simplement appuyer sur le bouton correspondant. Une telle représentation graphique de l'historique de la consultation est importante pour réduire la distance sémantique (car les boutons expriment directement les écrans auxquels ils font référence).



Figure 6-20. L'écran "Historique"

b. Les écrans avec informations terminales

Ce sont les écrans contenant les informations recherchées par l'utilisateur. Lorsque celui-ci est positionné sur ce type d'écran, il ne sait pas aller plus loin et est donc obligé de revenir en arrière ou d'arrêter sa consultation. Nous allons, comme pour les écrans de choix, présenter les différents types d'écrans avec informations terminales.

Le premier de ces écrans est représenté à la Figure 6-21. Il s'agit en fait de la présentation sous la forme d'un diaporama de la ville ou de l'université sélectionnée précédemment par l'utilisateur. Celui-ci n'a aucun pouvoir sur le déroulement de ce diaporama, il peut juste l'arrêter en activant le bouton graphique situé en bas à droite de l'écran. C'est la raison pour laquelle la zone de menu est vide. Pour permettre une lecture adéquate des informations textuelles, celles-ci apparaissent chronologiquement en trois zones. Certaines de ces informations sont complétées soit par un dessin ou par une photo. A la Figure 6-22, on trouve une autre possibilité pour la présentation de la ville et de l'université de Strasbourg. En effet, il nous avait été demandé d'insérer des films vidéos retraçant leurs histoires respectives. L'écran se compose donc d'un poste de télévision dans lequel le petit film défilera. On peut également

constater la présence d'un bouton graphique en bas à droite de l'écran. Celui-ci permet à l'utilisateur, comme pour le diaporama, de quitter la séquence vidéo s'il le désire.

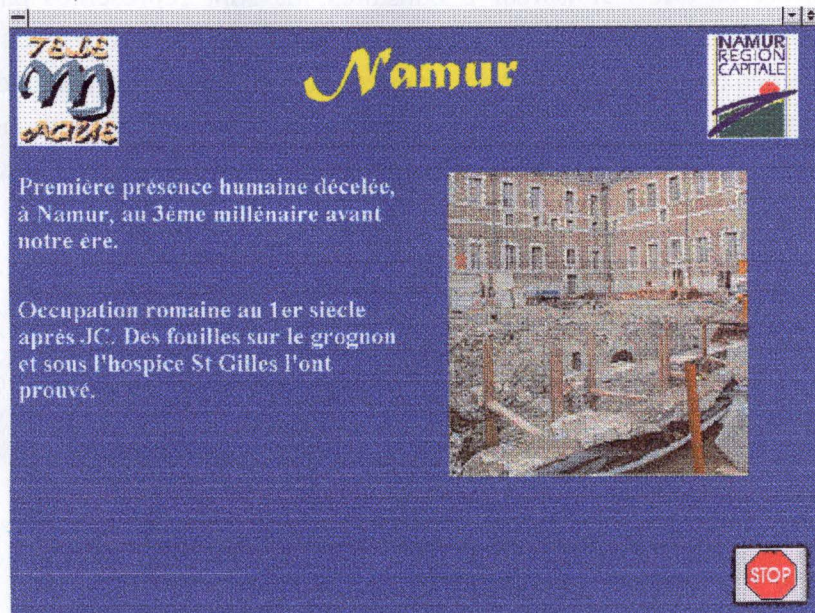


Figure 6-21. L'écran "Présentation de la ville"



Figure 6-22. L'écran "Présentation de Strasbourg"

Le troisième écran que l'on souhaite présenter est celui obtenu après avoir sélectionné un groupe de cours de l'écran représenté en Figure 6-17, il s'agit de l'écran reproduit à la Figure 6-23. Il nous permet de voir l'apparition d'une liste de sélection. Cet objet nous a semblé être, dans ce cas, le plus approprié vu que le nombre d'items à afficher est inconnu a priori et qu'il faut faciliter au maximum l'accès de l'utilisateur à l'information.



Figure 6-23. L'écran "Cours"

Le quatrième écran que nous présentons représente l'écran type des informations pratiques. On peut consulter cet écran à la Figure 6-24. Il se divise en quatre parties au plus. Chacune d'entre elle contient trois libellés. Chaque partie peut être utilisée pour définir un lieu ou une activité selon l'information recherchée. A la Figure 6-24, les informations reprises concernent les musées de la région namuroise. Dans chaque partie de texte, la première ligne donne le nom du musée, la deuxième fournit l'adresse et la troisième, une remarque complémentaire.

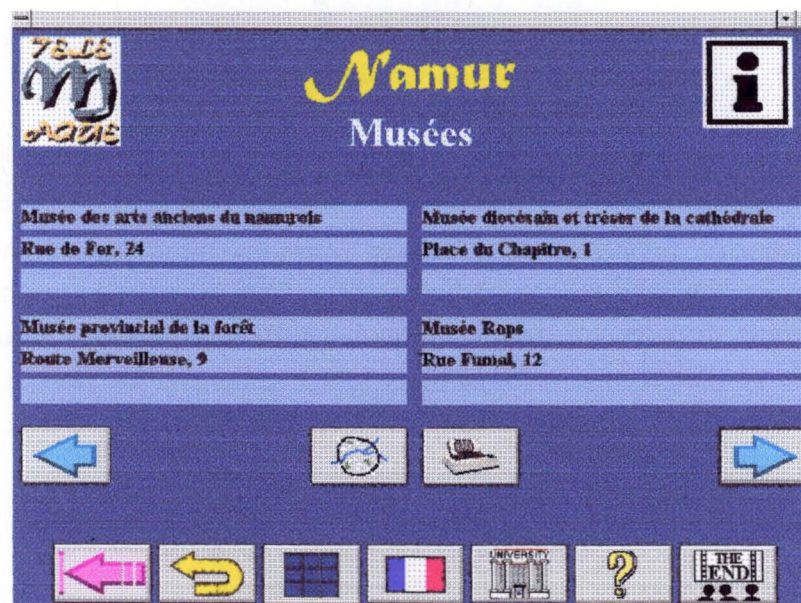


Figure 6-24. L'écran "Musées"

De plus, quatre nouveaux boutons graphiques font leur apparition. Tout d'abord, les flèches que l'on retrouve à la Figure 6-25. Elles vont permettre à l'utilisateur de voyager parmi les informations du type concerné s'il y en a plus de quatre.

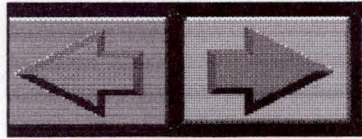


Figure 6-25. Les flèches

Le troisième bouton est celui de la Figure 6-26. En l'activant, l'utilisateur verra apparaître une carte de la ville où il pourra localiser les lieux identifiés sur un écran du type de la Figure 6-24.

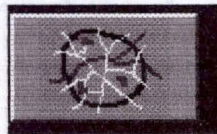


Figure 6-26. Plan de la ville

L'utilisateur retrouvera le dernier bouton (Figure 6-27), à chaque fois que des informations pourront être imprimées.

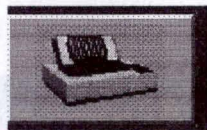


Figure 6-27. Impression

Le cinquième écran va permettre de présenter quelques informations chiffrées sur les universités. Il s'agira par exemple du nombre d'étudiants dans une faculté, du nombre d'étudiants par professeur ou encore de l'évolution du nombre d'étudiants dans une faculté. Les différents types de graphiques proposés seront les histogrammes, les aires et les lignes. Le graphique de la Figure 6-28 reprend un graphique de type histogramme avec l'évolution du nombre d'étudiants aux Facultés Universitaires de Namur.

Il faut noter que l'on peut retrouver sur ces écrans les flèches de la Figure 6-25. Celles-ci seront présentes lorsque plusieurs graphiques seront disponibles.

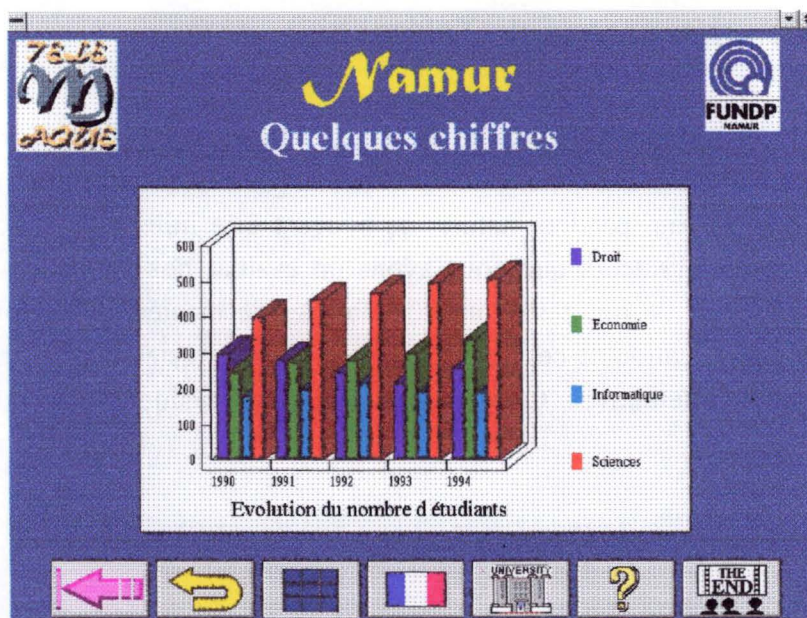


Figure 6-28. L'écran "Quelques chiffres"

c. Les écrans de transition

Les écrans de transition sont les écrans dans lesquels aucun choix ne doit être effectué et qui contiennent des informations non terminales. Nous n'avons qu'un seul écran de ce type que nous proposons à la Figure 6-29. Il permet de passer de la première série de choix (du choix initial de la langue jusqu'au choix de la ville) à la recherche d'informations plus précises. Cet écran possède deux zones de photos pour illustrer la ville choisie, un libellé d'information sur la ville et un bouton de commande situé sur la droite de l'écran possédant le libellé "OK". L'utilisateur devra l'actionner pour passer à la suite.



Figure 6-29. L'écran "Introduction de la ville"

6.3. Evaluation des écrans de Télémaque

Après avoir détaillé les écrans de Télémaque, nous allons maintenant les évaluer. Pour ce faire, nous utiliserons les critères d'évaluation de Shneiderman [SHN87]. Ils sont au nombre de six. Les critères de Shneiderman, par leur nature, ne peuvent être utilisés qu'a posteriori. C'est donc suite à nos tests utilisateurs que nous avons pu évaluer nos écrans au regard de ces critères. Malheureusement, si nos tests ont été effectués de manière assez formelle (le titre 7.2.5 est consacré à l'explication de ces tests), la population des testeurs était trop réduite pour en tirer des conclusions définitives. On peut provisoirement évaluer nos écrans de la façon suivante.

6.3.1. *Le temps d'apprentissage*

Vu le mode d'utilisation d'une borne, nous avons essayé de réduire le temps d'apprentissage au minimum. Selon les tests effectués, un utilisateur a compris le fonctionnement des principales fonctions du système en quelques secondes. Nous pouvons également dire que nous n'avons pas élaboré de guide de l'utilisateur. En effet, nous avons essayé de rendre l'utilisation de la borne assez simple et intuitive pour pouvoir s'en passer.

6.3.2. *La rapidité d'exécution d'une tâche*

La rapidité d'exécution d'une tâche peut être qualifiée de correcte avec Télémaque. Pour obtenir une information, l'utilisateur devra passer, en moyenne, par environ huit écrans intermédiaires. Mais ce nombre peut être réduit par l'utilisateur qui utilisera les fonctions d'historique ou de changement d'université disponibles dans la barre de menu graphique.

6.3.3. *Le taux d'erreurs effectuées par l'utilisateur*

Le taux d'erreurs est difficile à estimer vu le nombre peu élevé de tests (une douzaine) que nous avons pu réaliser. Mais nous pouvons dire, suivant nos observations, qu'elles étaient peu nombreuses et concernaient généralement la terminologie du classement des informations pratiques. De plus, l'utilisateur n'éprouvait aucun mal à corriger le tir.

6.3.4. *La période de rémanence*

La période de rémanence est la période pendant laquelle un utilisateur peut se souvenir des fonctions attachées à différents contrôles et du fonctionnement d'un logiciel en général. Nous n'avons pas effectué de tests pour mesurer la période de rémanence, mais les personnes qui ont pu tester Télémaque à différents moments n'éprouvaient aucune difficulté à se remémorer la signification des différents icônes. Nous pouvons donc considérer la rémanence comme bonne au niveau de l'utilisation de Télémaque.

6.3.5. La satisfaction subjective à utiliser le système

Les étudiants qui ont eu l'occasion de tester Télémaque étaient très satisfaits. Pour eux, la possibilité d'avoir tous ces renseignements sous la main et de manière aussi rapide était véritablement une aubaine. Le fait de pouvoir imprimer les informations recherchées amplifiait ce sentiment de satisfaction.

6.3.6. La couverture des commandes

Ce dernier critère de Shneiderman n'a pas beaucoup d'intérêt dans le cas de Télémaque. L'ensemble des commandes étant assez réduit, elles sont presque toutes utilisées de manière équivalente. Nous pourrions peut être quand même dire que la fonction historique était la moins utilisée, alors que nous avons trouvé son apport comme potentiellement important.

6.4. Gestion des différents média

Le logiciel que nous avons réalisé étant un logiciel multimédia, il convient de montrer comment les différents média présents dans Télémaque ont vu le jour et comment nous les avons intégrés dans le logiciel.

6.4.1. Le texte

Vu le caractère multilingue de notre borne, des traductions s'imposaient. Nous avons donc travaillé en collaboration avec des étudiants étrangers de Strasbourg. Cela posait comme problème que toute modification d'un texte devait leur être soumise pour traduction. Même si les réunions étaient assez fréquentes, la procédure était très lourde et cela nous faisait réfléchir à deux fois avant de changer quelque chose.

6.4.2. Le son

a. La voix

Il faut prendre ses précautions lors des enregistrements sinon la qualité pourra s'en ressentir. En effet, si le son n'est pas bon, l'intelligibilité du message va diminuer. Or, nous avons décidé d'utiliser la voix pour réaliser un système d'aide vocal, fonctionnalité où la compréhension est très importante. Il fallait donc essayer de travailler avec un micro sur pied, mettre un voile de nylon devant le micro pour couper le souffle, bien configurer la table de mixage du logiciel utilisé pour l'enregistrement... Par après, il y avait toujours moyen de retravailler le son avec des logiciels tels que l'enregistreur de sons sous Windows ou Wave Studio.

b. La musique

Nous avons décidé d'agrémenter d'un morceau de musique les présentations des villes et des universités. Nous devons donc choisir le format de fichier son qui allait être utilisé: les fichiers WAVE ou les fichiers MIDI. Nous allons expliquer pourquoi nous avons opté pour la deuxième solution.

La norme MIDI (Music Instrument Digital Interface) est un protocole de communication. Cette norme est composée d'un vocabulaire (pour les commandes). Les commandes permettent de transmettre quelle note est jouée, sa vitesse, l'action du pédalier... Elle gère 16 canaux sur lesquels on peut faire passer différents sons. [TAC95]

Les fichiers MIDI ne contiennent que des commandes alors que les fichiers WAVE englobent les sons en entier. Pour une même durée de musique, un fichier MIDI est 100 fois plus petit (une dizaine de Kilo octets au lieu d'un Méga octet). L'utilisation des fichiers MIDI nous a permis de gagner de l'espace disque.

Une deuxième raison de notre choix est le point de vue juridique. Un fichier WAVE provient de l'enregistrement d'un compact disque. Il est donc nécessaire, pour l'employer dans le domaine public, de demander les autorisations nécessaires aux autorités compétentes. Cela entraîne une énorme paperasserie et plus de 100.000 FB à payer. Les fichiers MIDI, eux, sont exempts de droits car ils donnent accès à leur partition qui peut être modifiée à volonté.

Un "inconvenient" des fichiers MIDI est qu'ils ne contiennent pas de parole. Cela ne nous dérangeait pas car nous voulions de la musique instrumentale pour soutenir les présentations. L'utilisation de paroles risquerait de perturber l'utilisateur qui doit déjà se concentrer sur du texte et des photos.

6.4.3. L'image fixe

a. Les graphes

Pour la rubrique "Quelques chiffres" de Télémaque, il s'agissait de faire apparaître des graphes reprenant des statistiques sur une université. La question qui se posait était: qui va réaliser les graphes ? Est-ce la personne qui remplit la base de données ou est-ce la borne elle-même ? Nous avons choisi la deuxième option car elle est plus souple. En effet, une personne entre les chiffres dans la base de données et c'est Télémaque qui construit le graphe. En cas de modification, il suffit de changer un chiffre dans la base de données et le graphe sera reconstruit à la prochaine utilisation de la borne.

b. Les photos

Les photos présentes dans le logiciel ont été scannées et retouchées avec Photo Styler.

c. Le logo

Pour la conception du logo de Télémaque, nous avons travaillé avec un étudiant des beaux-arts. Selon nos spécifications, il a réalisé un logo sur papier, logo que nous avons ensuite scanné et retouché (voir Figure 6-30).

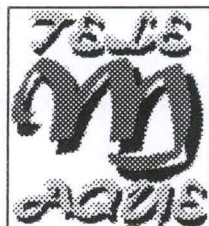


Figure 6-30. Logo de Télémaque

6.4.4. *L'image animée*

L'acquisition de la vidéo s'est faite grâce à la carte Vidéo Blaster RT300 et au logiciel Adobe Première.

6.4.5. *Intégration des média*

Cette section, se basant sur [KOE94], provient du fait que le terme multimédia implique l'intégration de différents média qui peuvent avoir des relations avec le temps. Ces relations temporelles sont une caractéristique du multimédia. Certains média tels que la vidéo et le son ont des exigences strictes de synchronisation temporelle. D'autres, comme le texte et la photo, ont des relations temporelles arbitraires.

La synchronisation, qui est la tâche de coordination des différents média, porte aussi bien sur la restitution de données de façon concurrente ou séquentielle que sur les événements externes générés par un utilisateur.

Les relations temporelles peuvent être implicites: la synchronisation est alors faite automatiquement. Par exemple, lors de l'acquisition d'une séquence vidéo, le son et l'image sont déjà synchronisés sur la cassette. Par contre, lorsqu'on ajoute un commentaire parlé à un texte, la synchronisation doit être explicite.

Nous allons, dans cette section, nous concentrer sur les média ayant des relations temporelles strictes (le texte et la photo peuvent devenir dépendants du temps s'ils sont animés selon le fonctionnement d'un diaporama) car ils sont les plus intéressants à étudier et nous montrerons comment nous avons appliqué les concepts dégagés dans la réalisation de notre borne.

a. Modèles conceptuels du temps

Il existe plusieurs manières de représenter le temps. Les deux plus courantes sont l'instant et l'intervalle. L'instant est un moment du temps de longueur zéro (par exemple, "16 heures") alors que l'intervalle a une durée définie entre deux instants (par exemple, de 13h à 17h).

Il y a 13 façons, pour deux intervalles, de se positionner dans le temps. La Figure 6-31 en présente 7; les 6 autres peuvent être trouvés en inversant les rôles de A et B (sauf pour la relation "égale").

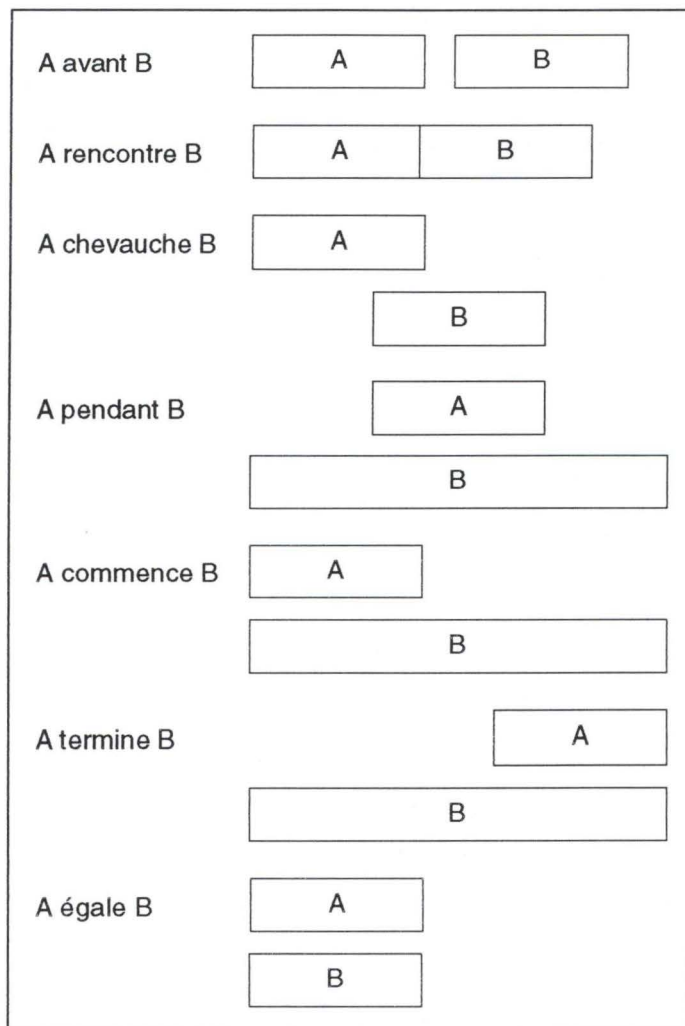


Figure 6-31. Relations temporelles binaires

Ces intervalles temporels peuvent être utilisés pour modéliser une présentation multimédia. Pour cela, il suffit d'attribuer à chaque intervalle un médium et une durée, et d'utiliser les 13 relations à notre disposition.

Dans Télémaque, la présentation des villes et des universités se fait de la manière suivante: du texte avec une photo associée apparaît toutes les x secondes, le tout soutenu par une musique

car leur apparition est conditionnée par le temps (le chronomètre). Nous sommes donc face à trois média dynamiques qui ont besoin d'être synchronisés. Nous pouvons, pour cela, utiliser la représentation par intervalles temporels et les relations temporelles binaires suivantes (voir Figure 6-32): "rencontre" (entre texte 1 et texte 2), "égale" (entre texte 2 et photo 2), "pendant" (entre photo 2 et musique 1), "commence" (entre texte 1 et musique 1), "termine" (entre photo n et musique 1).

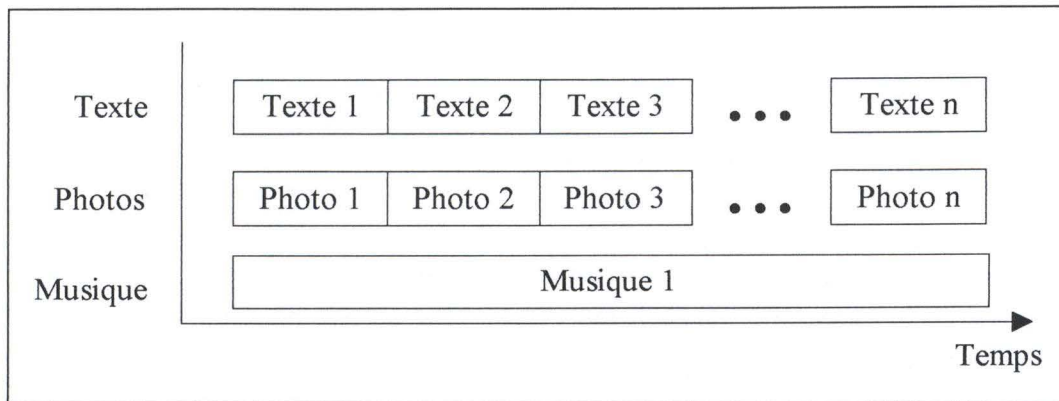


Figure 6-32. Représentation de la présentation des villes et des universités selon les intervalles temporels

b. Modèles conceptuels de la synchronisation

Après avoir vu comment représenter les informations par rapport au temps et les unes par rapport aux autres, nous allons maintenant aborder un autre problème: trouver un modèle conceptuel qui puisse décrire l'information temporelle nécessaire pour représenter la synchronisation multimédia.

La présentation des informations implique l'affichage simultané, séquentiel ou indépendant de données hétérogènes. Cela ressemble à l'exécution de processus séquentiels ou en parallèle, domaine dans lequel il existe de nombreux langages gérant la concurrence. La différence principale est que, dans le cas du multimédia, le but n'est pas la vitesse de résolution mais une présentation cohérente à l'utilisateur. Ce facteur humain entraîne une limitation dans la vitesse de restitution de l'information de façon à respecter sa capacité d'assimilation. Nous ne pouvons donc pas nous servir des modèles abstraits de représentation de la concurrence pour résoudre notre problème.

Dans la suite de cette section, nous allons identifier les exigences des données multimédia par rapport au temps et à la synchronisation.

(1) Contrôle d'accès temporel

La représentation des média doit permettre les opérations de contrôle d'accès temporel. Ces opérations fournissent les fonctionnalités de base sur lesquelles des logiciels multimédia relatifs au temps peuvent être construits. Les opérations suivantes sont des exemples d'opérations de contrôle d'accès temporel: lecture inverse, avance rapide, retour rapide, accès aléatoire... Télémaque ne permet pas ces contrôles d'accès temporel car les contenus du médium son ayant une durée très courte, il n'est pas nécessaire de pouvoir arrêter le médium ou revenir en arrière. Pour la vidéo et la présentation des villes, ces contrôles pourraient être implémentés dans une version ultérieure de Télémaque.

(2) Timing incomplet

Dans certaines circonstances, il peut être intéressant de ne pas spécifier complètement un timing. C'est le cas lorsqu'une donnée dynamique (dépendant du temps) est restituée en parallèle avec une donnée statique. Par exemple, une bande son est présentée en même temps qu'une photo. Celle-ci n'a pas une durée en elle-même puisque c'est une donnée statique. Dans ce cas, on ne spécifie pas le temps de présentation de ce médium et on suppose que la durée totale de la présentation est égale à la durée de restitution du médium dynamique (le son). Si les deux média sont dynamiques et ont des durées prédéterminées inégales, la synchronisation exige qu'un des deux média modifie sa durée de présentation.

Pour la présentation des villes et des universités dans Télémaque (voir Figure 6-32), il est intéressant d'avoir recours au timing incomplet. En effet, la durée de la présentation, qui dépend du nombre de couples texte/photo, va différer d'une ville (ou université) à l'autre. Dès lors, il est impossible de déterminer la durée du médium musique. Or, comme on vient de le dire, la synchronisation exige qu'un des deux modifie sa durée de présentation. Ici, lorsqu'il n'y a plus de couples texte/photo à présenter, le médium musique est arrêté.

(3) Transformations non temporelles

Ces opérations non temporelles, également appelées opérations spatiales, comprennent la diminution ("fading") des signaux vidéos et audios, le mixage des pistes, l'amélioration des couleurs des images, la génération de polices de caractères...

Pour représenter une orchestration multimédia, il est nécessaire d'avoir à la fois une représentation basée sur le temps et un modèle de transformations spatiales. Dans Télémaque, la présentation de la ville et de l'université de Strasbourg se fait au moyen de la vidéo. A partir de cassettes, nous avons introduit plusieurs séquences dans l'ordinateur. L'étape suivante était donc de relier ces séquences entre elles, étape réalisée grâce à une transformation non temporelle (spatiale): le "fading". Le logiciel Adobe Première emploie une représentation des média par intervalles temporels et permet les transformations spatiales. La Figure 6-33 nous montre deux séquences vidéos auxquelles on applique un "fading". En plus, Adobe Première propose différents types d'enchaînements pour rendre cette transformation spatiale aussi esthétique que possible.

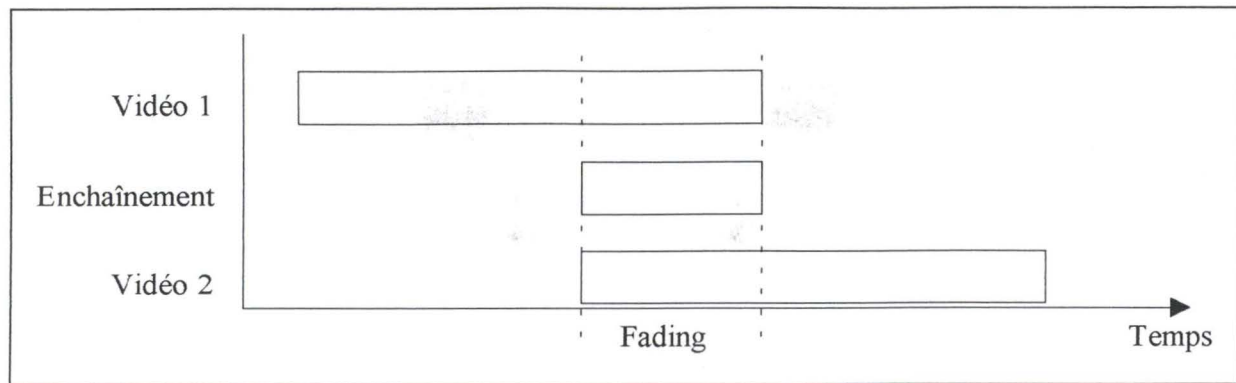


Figure 6-33. Transformation non temporelle au niveau de la vidéo

(4) Interaction et synchronisation

Quand un utilisateur interagit avec un logiciel multimédia, il faut qu'il y ait synchronisation entre cet utilisateur et le logiciel. L'utilisateur doit pouvoir démarrer ou arrêter la présentation d'un médium, interroger la base de données ou avoir n'importe quelle réaction permise par le système.

Imaginons une base de données reprenant des séquences vidéos de toutes les rues d'une ville et un logiciel permettant à l'utilisateur de se promener dans cette ville. Dans ce cas, la synchronisation est la coordination entre la présentation multimédia et des événements externes aléatoires (créés par l'utilisateur) car on ne sait pas, a priori, vers où l'utilisateur va vouloir aller. L'ordonnancement des séquences vidéos est donc dynamique (il ne peut pas être prédéterminé).

Pour la présentation de Strasbourg (au moyen de la vidéo), ce principe pourrait être appliqué et la vidéo deviendrait donc interactive: l'utilisateur pourrait se promener dans Strasbourg en choisissant lui-même l'itinéraire.

c. Modèle de visualisation

Le développeur a besoin, lorsqu'il travaille sur une présentation multimédia, d'une abstraction ou visualisation du produit final. Des schémas de représentation sont conçus pour cela et sont basés sur un graphe ou sur un langage, le but étant d'identifier les impératifs de concurrence et de timing.

L'avantage de la représentation basée sur les graphes est son côté visuel: le développeur peut facilement voir quelle est l'orchestration prévue. Il existe différents graphes sur lesquels peut reposer ce type de représentation. La Figure 6-34 présente l'orchestration de la présentation des villes et des universités selon un réseau de Pétri. Celui-ci permet la capture des relations temporelles. Dans notre exemple, relativement simple, une place représente la restitution d'un objet multimédia et les transitions, des points de synchronisation.

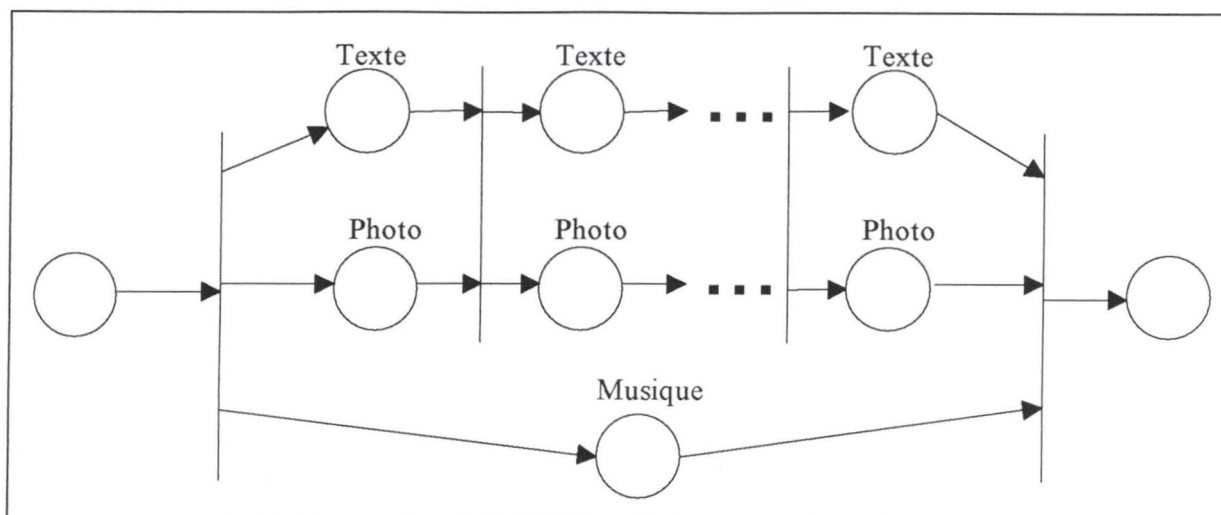


Figure 6-34. Utilisation d'un réseau de Pétri pour représenter la présentation des villes et des universités

La représentation basée sur les langages n'a pas le côté visuel des graphes mais elle gagne en puissance. En effet, tout formalisme basé sur des graphes est limité aux objets de ces graphes. Un langage peut donc dire plus de choses et peut même être utilisé conjointement à un graphe.

Par exemple, pour exprimer que A a lieu avant B, on écrit $A \gg B$. Maintenant, si on veut dire que A se produit en même temps que B, on écrit $A \parallel B$.

Ce chapitre clôture la présentation des deux points essentiels issus de notre méthodologie: le scénario et les interfaces. Nous avons d'abord rappelé quelques notions d'ergonomie. Celles-ci nous ont permis de caractériser les différents composants de Télémaque. Ensuite, nous sommes passés à l'étude des écrans de notre borne interactive. Dans ce point, nous avons présenté les différents moyens d'interaction ainsi que les principaux objets abstraits que nous utilisons dans Télémaque. Nous avons également défini les critères de conception sur lesquels nous nous sommes basés pour réaliser nos écrans, et nous avons montré comment nous avons tenu compte de ces critères de conception. Après cela, nous avons proposé une liste de critères ergonomiques orientés vers les bornes interactives multimédia. Nous avons ensuite décrit les écrans de Télémaque. Dans une première phase, nous avons montré leurs points communs, puis nous avons passé en revue les écrans classés par type. Enfin, nous avons réalisé une brève évaluation des écrans de Télémaque. Pour terminer ce chapitre, nous avons présenté la gestion des différents média. Dans le chapitre suivant, nous nous proposons d'énumérer et de détailler les différentes étapes de la conception de Télémaque.

7. Réalisation de la borne

Ce chapitre explique les différentes étapes qui ont été effectuées pour réaliser la borne Télémaque. Dans la section 7.1, nous justifions le choix de notre plate-forme de développement matérielle et logicielle. La section 7.2 reprend les différentes étapes de la méthodologie retenue et explique leur contenu dans le cadre de la réalisation de notre borne.

7.1. La plate-forme de développement

Cette section se divise en deux parties. D'une part, nous expliquons les choix que nous avons effectués au niveau du matériel (caractéristiques de la machine de développement et de la machine cible) et d'autre part, nous justifions nos choix en ce qui concerne les langages de développement et de prototypage, et le gestionnaire de base de données.

7.1.1. Choix de l'environnement matériel

Il y a quelques années, lorsque le mot multimédia était prononcé, on pensait quasi immédiatement "Macintosh". En effet, le constructeur Apple avait réussi à imposer sur les marchés l'architecture multimédia de ses ordinateurs. Bien entendu, au départ, les configurations étaient un peu limitées, mais tout était déjà intégré: carte son d'abord, puis cartes vidéo; on voit même apparaître des extensions pour la conception professionnelle de 3D.

Pour notre part cependant, la plate-forme P.C. a été imposée par le client.

La configuration de la machine de développement et de ces périphériques était la suivante.

- ◇ La machine est un IBM 486 DX2/66.
- ◇ Elle possède un clavier étendu et une souris.
- ◇ Sa mémoire vive est de 32 Mégas.
- ◇ La mémoire secondaire est composée de deux disques durs de 350 Mégas chacun. Il fallait en effet une capacité de mémoire secondaire assez grande pour installer les différents logiciels qui nous ont permis de développer la borne.
- ◇ L'écran tactile est de marque NEC. Nous avons choisi de prendre un écran de ce type parce que ses performances étaient meilleures que les écrans IBM. Nous avons remarqué, par expérience, qu'il présente non seulement une meilleure sensibilité au toucher mais aussi des modes de fonctionnement plus complets: il peut fonctionner en mode souris, en mode tactile ou en mode mixte, et on peut paramétrer la présence ou non du curseur et le feed-back sonore d'une sélection via l'écran tactile.

- ◇ Elle comprend une carte son AWE32 de Creative Labs. Cette carte son est probablement trop évoluée pour l'utilisation qui en est faite, mais elle a une caractéristique très intéressante: elle permet de jouer des fichiers musicaux MIDI avec une très bonne qualité de restitution, ce que ne permettent généralement pas les autres cartes son.
- ◇ Elle possède une carte de "capture" vidéo RT300 également de Creative Labs. Celle-ci nous a permis de "capturer" les séquences vidéos présentes dans la borne. Le choix de cette carte a été effectué après une petite étude de marché et quelques tests. La RT300 rassemblait les principaux critères que nous recherchions: elle permet la compression en temps réel des séquences vidéos qu'elle "capture", ce qui élimine la phase de compression après coup et réduit donc le temps global nécessaire à l'acquisition d'une séquence. De plus, elle était livrée avec le Runtime de Vidéo for Windows (indispensable pour la lecture simple des séquences vidéos) et avec le logiciel Adobe Première, qui permet de gérer les "captures" vidéos et les montages.

La machine cible varie très peu de la première machine de programmation. En voici les principales caractéristiques.

- ◇ Processeur 486 DX2/66.
- ◇ 32 Mégas de RAM.
- ◇ Disque dur de 2 ou 3 Gigas.
- ◇ Ecran tactile NEC.
- ◇ Clavier étendu et souris.
- ◇ Carte son AWE32 (Creative Labs).

7.1.2. Choix de l'environnement logiciel

a. Choix du langage de développement

Sur le marché, on trouve trois groupes de logiciels permettant de réaliser un logiciel multimédia: les logiciels de présentation multimédia, les systèmes auteurs et les langages de programmation. Vu l'évolution que connaissent ces trois catégories, la définition de leurs frontières est de plus en plus difficile à fixer. [KOE94]

(1) Les logiciels de présentation

La plupart des logiciels de présentation ne demandent pas de programmation. On a à sa disposition des "diapositives" dans lesquelles on peut insérer les différents média; ensuite, le logiciel de présentation ordonne ces "diapositives" en séquence. Ceci convient lorsque le logiciel à réaliser ne doit pas être interactif et qu'on veut obtenir très vite un produit fini.

A l'heure actuelle, on peut trouver des logiciels de présentation permettant une certaine interactivité. Le produit fini ressemble alors de plus en plus à ce qui peut être fait avec des systèmes auteurs.

Power Point et Asymetrix sont deux représentants de cette catégorie de logiciels.

(2) Les systèmes auteurs

Les systèmes auteurs permettent à des personnes non spécialistes en informatique d'utiliser la puissance de l'ordinateur pour fournir, à un utilisateur final, une expérience interactive (par exemple, une formation à une activité). Cela est rendu possible par la relative simplicité de programmation de ces systèmes (programmation souvent visuelle).

Ces systèmes ont pour but de fournir les outils nécessaires pour créer un logiciel interactif. Ils sont typiquement utilisés pour l'éducation, la formation professionnelle... Les "développeurs" sont ici des spécialistes dans le secteur d'activité pour lequel le logiciel est réalisé.

Pour les logiciels concernant l'enseignement, les systèmes auteurs sont particulièrement bien adaptés car ils permettent généralement de garder les traces d'une utilisation. Ainsi, après le passage d'un élève, le professeur peut voir à quelle leçon l'élève est arrivé et quel résultat il a obtenu. Cela n'est pas réalisable avec les logiciels de présentation.

Malheureusement, ces systèmes possèdent souvent quelques points faibles: « *ils présentent des limites dues à leur langage de scripting trop peu puissant, au manque de contrôle qu'ils permettent sur le hardware et le système d'exploitation (...), au manque d'ouverture sur le monde extérieur (manque de dialogue avec d'autres applications, données externes peu accessibles)* » [GOF95]. Par exemple, certains systèmes auteurs ne possèdent pas de gestionnaire de base de données (Authorware), d'autres offrent des possibilités d'interactivité assez limitées (MacroMind Director)...

Les systèmes auteurs les plus connus sont Toolbook, MacroMind Director, Authorware et IconAuthor.

(3) Les langages de programmation

Les langages de programmation demandent plus de connaissances informatiques et plus de travail que les deux premiers mais sont plus puissants, moins restrictifs. En effet, puisque la programmation retrouve ici une place prépondérante, les principales limitations dépendront

généralement davantage des potentialités des programmeurs que du langage de programmation lui-même.

On peut également profiter de la petite facilité de conception qu'offrent certains langages de programmation visuels. Ceux-ci permettent en effet de représenter graphiquement des morceaux de programme parfois très abstraits [GOF95]. Visual Basic et Visual C++ sont deux exemples de langages de programmation visuels.

(4) Notre choix

Notre choix doit porter, étant donné le problème, sur un langage de développement

- qui n'est pas en phase de naissance, pour minimiser les risques de développement;
- qui peut nous être enseigné (existence d'un consultant extérieur) et qui ne demande pas un trop long apprentissage, pour minimiser les risques de délai;
- qui peut gérer une grosse base de données, étant donné le problème posé;
- qui permet une interactivité importante;
- qui fonctionne sur P.C.;
- qui offre des primitives multimédia.

A la lumière de ces exigences, nous avons comparé un certain nombre de langages de développement. On peut consulter les résultats de notre comparaison au Tableau 7-1.

	Nature	Avantages	Inconvénients
Asymetrix	logiciel de présentation	☺ apprentissage rapide ☺ développement rapide	☹ interactivité limitée ☹ ne permet pas la gestion d'une base de données
Toolbook	système auteur (carte)	☺ gestion aisée des objets multimédia ☺ possibilité d'avoir recours à un consultant extérieur	☹ surtout pour la conception de logiciels EAO ☹ verbosité du langage ☹ apprentissage nécessaire
Authorware	système auteur (diagramme)	☺ facilité de programmation ☺ facilité de maintenance ☺ possibilité d'avoir recours à un consultant extérieur développement rapide car pas de langage	☹ pas de technologie pour gérer la base de données ☹ ne permet pas les impressions ☹ apprentissage nécessaire

IconAuthor	système auteur (diagramme)	☺ développement rapide car pas de langage	☹ apprentissage nécessaire ☹ pas de consultant extérieur
Windev	groupware avec langage de programmation	☺ contient tous les outils: 4GL, IHM, GBD, GMM...	☹ guidance trop forte ☹ apprentissage nécessaire ☹ pas de consultant extérieur ☹ technologie naissante
Power Builder	langage de programmation		☹ apprentissage nécessaire ☹ pas de consultant extérieur ☹ technologie naissante
Visual Basic Pro	langage de programmation	☺ spécialisé dans le multimédia ☺ permet de nombreuses extensions grâce à des librairies (VBX) ☺ possibilité d'avoir recours à un consultant extérieur ☺ technologie mûre	☹ apprentissage nécessaire ☹ langage semi-compilé (librairies)

Tableau 7-1. Etude du langage de développement

Au terme de cette étude restreinte des diverses possibilités, nous avons décidé de choisir Visual Basic Pro comme langage de développement de la borne interactive car il constitue une technologie mûre qui peut nous être enseignée assez rapidement. De plus, il peut gérer une grosse base de données et des impressions d'informations, il offre des primitives multimédia et permet de nombreuses extensions grâce à des librairies.

b. Choix du système de gestion de base de données

Il est difficile de parler ici de choix. En effet, Visual Basic possède des liens forts avec le système de gestion de base de données Access 2.0. Nous avons donc opté pour celui-ci qui se base sur le modèle relationnel.

c. Choix du langage de prototypage

Comme nous l'avons dit dans le chapitre consacré à l'explicitation de la méthodologie retenue pour Télémaque (chapitre 4), notre prototype est représenté par le scénario du logiciel. Ce scénario est une suite d'écrans avec une définition de la navigation qui est permise entre ces écrans. Cette présentation n'est pas interactive c'est pourquoi nous avons parlé de prototype non-opérationnel.

Dans ces conditions, un logiciel de présentation multimédia, tel que défini au point 7.1.2.a, convient parfaitement pour présenter les écrans au client vu la rapidité de développement et l'absence de programmation. Parmi les logiciels disponibles, nous avons choisi Power Point 4.0.

7.2. Les différentes étapes de la conception

La méthodologie que nous avons explicitée à la section 4.2 comportait les étapes suivantes: réalisation du prototype non-opérationnel, simulations rapides, démonstrations rapides et tests supplémentaires. Nous allons maintenant expliquer ce que contiennent ces différents points pour le développement de la borne Télémaque. Avant cela, nous évaluerons brièvement les risques liés à notre projet.

Nous voulons justifier ici pourquoi nous ne présenterons pas d'architecture de notre logiciel. Cela vient du fait que Visual Basic offre une programmation événementielle. Il comprend des contrôles et des événements. Les contrôles sont la fenêtre, le bouton, l'image, l'étiquette, le contrôle du multimédia, l'horloge... Les événements sont l'ouverture, la fermeture, l'activation, le click, le double-click... A chaque événement de chaque contrôle est associé une procédure qui est exécutée lorsque l'événement se produit. Dans d'autres langages tels que Cobol et Pascal, on peut identifier une procédure principale faisant appel à des routines. Ce n'est pas le cas dans Visual Basic.

7.2.1. *Evaluation des risques*

Pour évaluer les risques liés à notre projet, nous allons utiliser le classement des risques de Boehm (modèle de la U.S. Air Force).

Si l'on se reporte aux grilles d'évaluation des risques liés à un projet, on constate que, dans notre cas, le risque principal est technique. On peut en effet identifier

- *la disponibilité de la technologie* car nos machines de développement, la carte vidéo et le module pour interfacer Visual Basic et Access ne sont pas disponibles tout de suite;
- *la maturité de la technologie* car la carte vidéo que nous avons commandée n'a pas encore fait ses preuves;
- *l'expérience de l'équipe technique* car nous devons choisir un langage de développement et un gestionnaire de bases de données capables de traiter du multimédia; or notre expérience dans ce domaine est faible voire nulle.

Pour minimiser ces risques, nous avons décidé de faire appel à un consultant extérieur pour nous donner une formation au langage de développement que nous avons choisi (Visual Basic Pro) et pour répondre à nos questions pendant toute la réalisation du logiciel.

Un autre risque capital est celui de *la stabilité des spécifications*. En effet, celles-ci ne sont pas précises et sont souvent modifiées. Nous avons donc décidé de les geler à un certain moment. A partir de là, toute nouvelle demande du client sera enregistrée mais ne sera pas prise en compte immédiatement. Ce n'est qu'à la fin du projet que nous verrons s'il est possible de l'intégrer facilement dans le produit.

7.2.2. Réalisation du prototype non-opérationnel

Notre prototype non-opérationnel est matérialisé par le scénario de Télémaque. Comme nous lui avons déjà consacré une section (voir 5.3.3), nous ne reviendrons plus sur ce point.

7.2.3. Simulations rapides: développement des interfaces

a. Implantation des interfaces

A partir du scénario validé par le client, nous avons développé les écrans en Visual Basic. Nous y avons ensuite inséré des informations sur Namur afin de pouvoir procéder à des simulations rapides. Même si toutes les fonctions n'étaient pas encore disponibles, ces premiers tests utilisateurs ont servi à corriger ou à améliorer nos écrans. Voici quelques exemples de faits observés lors de ces tests et les modifications que nous avons apportées au programme pour remédier à ces "problèmes".

- Sur le premier écran (choix de la langue), les utilisateurs avaient tendance à appuyer sur les trois logos qui bougent et étaient étonnés de ne pas avoir de réactions de la part du logiciel. Nous avons donc décidé d'associer du son à chaque logo. Par exemple, lorsqu'on appuie sur le logo FUNDP, une voix vous explique la signification de ce sigle.
- Lorsque l'utilisateur arrivait au premier menu (menu principal), il n'identifiait pas la zone centrale de l'écran comme une zone de choix mais comme une zone d'information. Il n'avait donc pas tendance à appuyer sur les deux boutons présents dans cette zone. C'est pourquoi nous avons ajouté une flèche devant chaque bouton pour inciter l'utilisateur à appuyer sur ces boutons.

Les étapes de conception des écrans ainsi que d'acquisition et d'intégration des différents média ont été longuement développées au chapitre précédent. Nous nous contenterons donc de citer quelques difficultés liées à la réalisation de ces étapes.

b. Difficultés rencontrées

Lors de l'implantation des interfaces, nous avons rencontré des difficultés et avons dû faire des choix. Ce sont ces difficultés et ces choix (provenant du langage de développement lui-même ou des fonctionnalités de notre logiciel) que nous expliquons dans la suite de cette section.

(1) Difficultés provenant du langage de développement

Les objets Visual Basic mis à la disposition du créateur d'interfaces sont très classiques. La conception d'interfaces personnalisées et de qualité n'est donc pas aisée. Quand nous parlons de qualité, nous ne parlons pas du respect des critères ergonomiques mais plutôt du côté attrayant de la borne. Par exemple, il est impossible de mettre un dégradé en fond d'écran, de changer la couleur des boutons de commande (gris classique)...

Visual Basic pose également un problème au niveau du développement de logiciels. Le travail de plusieurs personnes sur un même projet est rendu difficile par la structure du langage (un seul fichier global).

Toutes les photos insérées dans le programme doivent avoir le même format (RGB). Si ce n'est pas le cas, certaines photos voient leurs couleurs altérées à cause de conflits au niveau des palettes de couleurs.

Enfin pour pouvoir profiter d'une barre d'icônes unique pour tous les écrans, nous avons utilisé une MDI Form (fenêtre restant constamment à l'écran sur laquelle les autres fenêtres viennent se greffer). Malheureusement, il n'est pas possible de lui retirer sa barre de titre (boîte de contrôle et boutons pour minimiser/maximiser la fenêtre). Cela nuit quelque peu à l'esthétique de nos écrans et cela nous a obligé à prendre des précautions pour empêcher un utilisateur de sortir du programme et d'aller naviguer dans d'autres logiciels.

(2) Difficultés provenant de la fonction changement de langue

Le changement de langue est une fonction importante du logiciel: il faut que celui-ci se fasse immédiatement et très rapidement. Nous ne voulons pas que l'utilisateur doive attendre d'être passé à l'écran suivant pour voir les informations dans la langue désirée.

Pour obtenir cette rapidité, nous avons décidé d'associer un tableau à chaque contrôle de chaque fenêtre. Ces tableaux ont chacun 5 éléments (5 langues pour l'instant) reprenant le libellé dans chaque langue. Une variable globale de langue est initialisée, au départ, par le choix de l'utilisateur et peut évoluer au cours de la consultation si l'utilisateur décide de changer de langue. Lorsque ce changement a lieu, les libellés des contrôles de la fenêtre active sont immédiatement modifiés. En général, les libellés d'une fenêtre sont initialisés, en fonction de la valeur de la variable globale de langue, chaque fois que cette fenêtre devient active.

(3) Difficultés provenant du chargement des fenêtres

Le chargement des fenêtres posait un problème: trois possibilités s'offraient en fait à nous:

1. le chargement d'une fenêtre en mémoire lorsqu'elle est appelée et le déchargement dès qu'on passe à une autre fenêtre, ce qui a pour résultat une optimisation de la mémoire mais pas du temps de réponse;

2. le chargement de toutes les fenêtres au début et le déchargement de toutes les fenêtres à la fin, ce qui a pour résultat une optimisation du temps de réponse mais pas de la mémoire;
3. le chargement de certaines fenêtres au début (les plus utilisées) et des autres en cours d'utilisation ainsi que le déchargement de certaines fenêtres (les moins utilisées) dès qu'on les quitte et des autres à la fin, ce qui représente un juste milieu.

Vu la configuration de la machine cible (32 Mégas de mémoire) et le nombre total de fenêtres, il nous semble opportun de privilégier le temps de réponse par rapport à une utilisation optimale de la mémoire. Les fenêtres seront donc toutes chargées au démarrage du logiciel. Le temps d'attente est concentré à cet endroit et, vu l'utilisation faite de notre logiciel (il ne sera lancé qu'une fois par jour puis tournera toute la journée en continu), ce choix paraît être le plus pertinent.

(4) Difficultés provenant du système d'aide

Un développeur de logiciels doit se poser la question suivante: « Pourquoi les utilisateurs n'utiliseraient-ils pas mon système d'aide ? ». A cela, plusieurs réponses sont possibles: ils se plaignent de

- la difficulté de trouver l'information;
- la difficulté de passer du contexte de travail au système d'aide;
- la complexité de l'interface du système d'aide;
- la qualité de l'information.

Le développeur doit donc anticiper les besoins des futurs utilisateurs pour éviter qu'une de ces quatre critiques ne puisse être émise.

Une question que nous nous sommes posés est: sur quoi va porter notre système d'aide ?

Sur la signification de la barre d'icônes ou sur l'utilisation générale du logiciel ? Nous avons pour cela passé en revue différents types de système d'aide reposant chacun sur une philosophie différente. [LAU90]

- Goal-oriented questions
Que puis-je faire avec ce programme ?
- Descriptive questions
Qu'est-ce que ceci et à quoi ça sert ? L'utilisateur doit être capable de désigner les objets à propos desquels il veut une explication.
- Procedural questions
Comment puis-je faire ceci ?

- Interpretive questions
Pourquoi ceci s'est-il passé ? L'utilisateur ne comprend pas pourquoi le système a répondu de telle manière à son action.
- Navigational questions
Où suis-je ?

Une aide sur l'ensemble du logiciel ne nous semblant pas indispensable, nous nous sommes limités à expliquer notre menu graphique. En effet, l'utilisation se résumant à une simple navigation, cela ne présente pas un effort de compréhension trop important. Par contre, la signification des icônes n'est pas toujours évidente et nécessite quelques explications. Nous avons donc réalisé un système d'aide du type "descriptive questions". Lorsque le système d'aide est actif, le fait d'appuyer sur un bouton (l'utilisateur est capable de désigner l'objet à propos duquel il veut une explication) déclenche l'explication orale de sa fonction (à quoi ça sert).

En plus, nous avons ajouté une aide du type "navigational questions". Grâce aux libellés et logos d'un écran, l'utilisateur peut savoir dans quelle partie du logiciel il se trouve.

7.2.4. Démonstrations rapides: développement de la base de données

Avant de pouvoir procéder à des démonstrations rapides, nous avons dû développer la base de données. Il a fallu, pour cela, élaborer le schéma conceptuel et le transformer pour qu'il respecte le modèle relationnel, créer et remplir la base de données à l'aide d'Access et finalement, modifier le code du programme obtenu jusque là.

a. Réalisation des différents schémas

(1) Le schéma conceptuel

Notre schéma conceptuel (voir Figure 7-1) contient les entités suivantes:

- pays: pays dans lequel se trouve l'université partenaire;
- ville: ville dans laquelle se trouve l'université partenaire;
- presville: présentation de la ville;
- infopratt: informations pratiques concernant la ville et l'université;
- université: université dans laquelle un étudiant peut faire son stage;
- presuniv: présentation de l'université;
- graphique: statistiques sur l'université;
- chiffre: valeur des statistiques;
- faculté: faculté, au sein de l'université, dans laquelle un étudiant peut faire son stage;
- organisation: organisation de la faculté;
- cours: cours existant dans la faculté.

Le Tableau 7-2 présente les identifiants de chaque entité.

Entité	Identifiant
PAYS	Nom_P
VILLE	Nom_V + PAYS
PRESVILLE	N°_Pres + VILLE
INFOPRAT	Type_Inf + Nomfra_Inf + VILLE
UNIVERSITE	Nom_U + VILLE
PRESUNIV	N°_Presu + UNIVERSITE
GRAPHIQUE	Nomfra_G + UNIVERSITE
CHIFFRE	Champfra_G + GRAPHIQUE
FACULTE	Nom_F + UNIVERSITE
ORGANISATION	Nom_O + Foncfra_O + FACULTE
COURS	Annee + Intitulefra + FACULTE

Tableau 7-2. Identifiants du schéma conceptuel

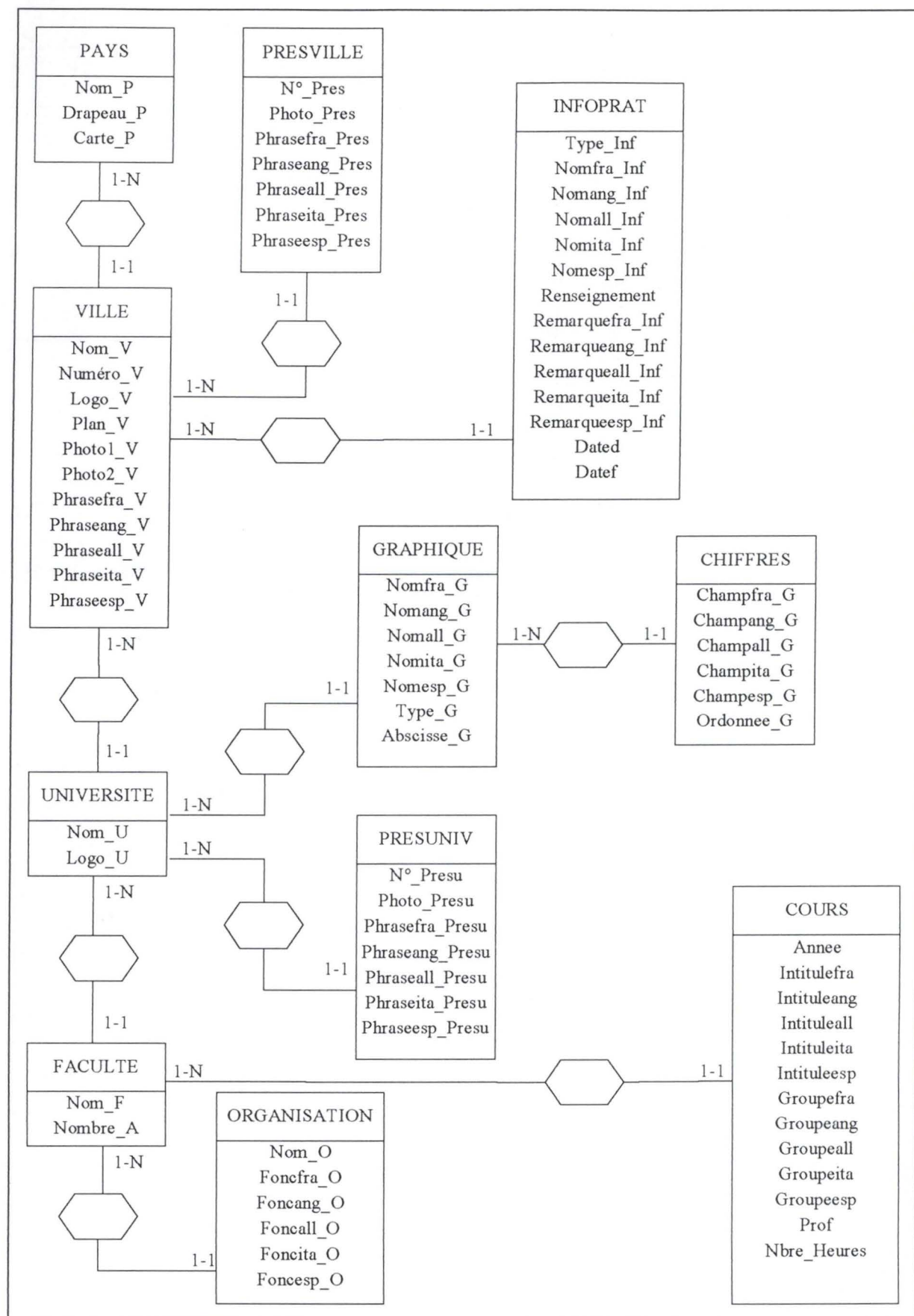


Figure 7-1. Le schéma conceptuel

(2) Le schéma conforme au modèle relationnel

Comme Access est un gestionnaire de bases de données relationnel, il faut transformer notre schéma conceptuel et le rendre conforme au modèle relationnel.

Dans notre schéma transformé (voir Figure 7-2), les identifiants des tables apparaissent en gras italique.

Cette transformation entraîne l'apparition de contraintes référentielles de deux types: celles exprimant les clés externes ("foreign key") et celles exprimant la connectivité minimale 1 dans notre schéma conceptuel.

Voici ces contraintes référentielles.

Exprimant la clé externe

- Tout *NOM_P* de *VILLE* est un *NOM_P* de *PAYS*.
- Tout (*NOM_P*, *NOM_V*) de *UNIVERSITE* est un (*NOM_P*, *NOM_V*) de *VILLE*.
- Tout (*NOM_P*, *NOM_V*, *NOM_U*) de *FACULTE* est un (*NOM_P*, *NOM_V*, *NOM_U*) de *UNIVERSITE*.
- Tout (*NOM_P*, *NOM_V*) de *INFOPRAT* est un (*NOM_P*, *NOM_V*) de *VILLE*.
- Tout (*NOM_P*, *NOM_V*) de *PRESVILLE* est un (*NOM_P*, *NOM_V*) de *VILLE*.
- Tout (*NOM_P*, *NOM_V*, *NOM_U*) de *PRESUNIV* est un (*NOM_P*, *NOM_V*, *NOM_U*) de *UNIVERSITE*.
- Tout (*NOM_P*, *NOM_V*, *NOM_U*) de *GRAPHIQUE* est un (*NOM_P*, *NOM_V*, *NOM_U*) de *UNIVERSITE*.
- Tout (*NOM_P*, *NOM_V*, *NOM_U*, *NOMFRA_G*) de *CHIFFRES* est un (*NOM_P*, *NOM_V*, *NOM_U*, *NOMFRA_G*) de *GRAPHIQUE*.
- Tout (*NOM_P*, *NOM_V*, *NOM_U*, *NOM_F*) de *COURS* est un (*NOM_P*, *NOM_V*, *NOM_U*, *NOM_F*) de *FACULTE*.
- Tout (*NOM_P*, *NOM_V*, *NOM_U*, *NOM_F*) de *ORGANISATION* est un (*NOM_P*, *NOM_V*, *NOM_U*, *NOM_F*) de *FACULTE*.

Exprimant la connectivité minimale 1

- ♦ Tout NOM_P de PAYS est un NOM_P de VILLE.
- ♦ Tout (NOM_P, NOM_V) de VILLE est un (NOM_P, NOM_V) de UNIVERSITE.
- ♦ Tout (NOM_P, NOM_V) de VILLE est un (NOM_P, NOM_V) de PRESVILLE.
- ♦ Tout (NOM_P, NOM_V) de VILLE est un (NOM_P, NOM_V) de INFOPRAT.
- ♦ Tout (NOM_P, NOM_V, NOM_U) de UNIVERSITE est un (NOM_P, NOM_V, NOM_U) de FACULTE.
- ♦ Tout (NOM_P, NOM_V, NOM_U) de UNIVERSITE est un (NOM_P, NOM_V, NOM_U) de PRESUNIV.
- ♦ Tout (NOM_P, NOM_V, NOM_U) de UNIVERSITE est un (NOM_P, NOM_V, NOM_U) de GRAPHIQUE.
- ♦ Tout (NOM_P, NOM_V, NOM_U, NOM_F) de FACULTE est un (NOM_P, NOM_V, NOM_U, NOM_F) de ORGANISATION.
- ♦ Tout (NOM_P, NOM_V, NOM_U, NOM_F) de FACULTE est un (NOM_P, NOM_V, NOM_U, NOM_F) de COURS.
- ♦ Tout (NOM_P, NOM_V, NOM_U, NOMFRA_G) de GRAPHIQUE est un (NOM_P, NOM_V, NOM_U, NOMFRA_G) de CHIFFRES.

D'où le nouveau schéma suivant conforme au modèle relationnel.

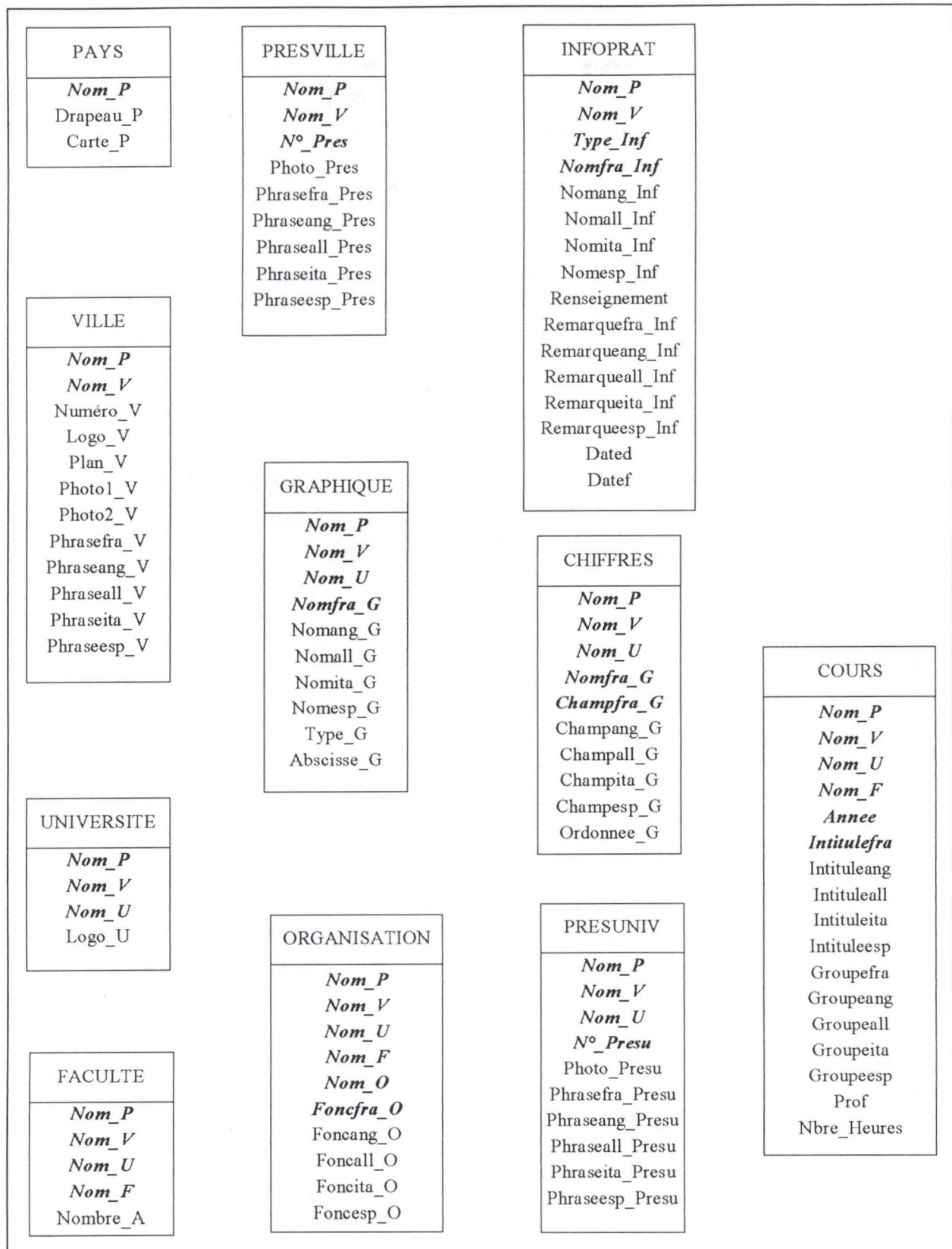


Figure 7-2. Schéma conforme au modèle relationnel

b. Implantation de la base de données

Cette étape consiste à créer la base de données à l'aide du gestionnaire de base de données Access, à concevoir les formulaires pour permettre l'insertion d'informations et à commencer à remplir la base de données.

(1)Création des tables et des relations

Access étant un gestionnaire de base de données basé sur le modèle relationnel, il était facile de créer la base de données à partir du schéma conforme au modèle relationnel (voir Figure 7-2).

(2)Création des formulaires

Comme nous l'avons dit dans le cahier des charges (voir 3.2.2.c), la base de données doit pouvoir être mise à jour par des non-spécialistes en informatique. Access permet de réaliser des formulaires c'est-à-dire des écrans liés aux tables de la base de données. Ceci facilite le remplissage car il s'effectue dans ces formulaires (respectant les critères de [VAN94]) et non dans des tableaux difficilement compréhensibles par des novices.

(3)Remplissage de la base de données

Nous avons, dans un premier temps, introduit les informations sur Namur et regardé si le programme se comportait de la même façon que précédemment (quand les informations ne se trouvaient pas dans la base de données). Ensuite, nous avons pu compléter la base de données avec des informations sur Strasbourg.

c. Modification du code

Nous avons dû modifier le code du programme en Visual Basic en y intégrant des requêtes SQL pour que le logiciel vienne chercher les informations dans la base de données. Jusque maintenant, les informations étaient contenues dans le logiciel lui-même et la recherche ne nécessitait donc pas de réelles requêtes. Nous avons utilisé le langage d'interrogation SQL vu que notre gestionnaire de base de données est relationnel.

Lorsque le code du programme a été modifié, nous avons pu réaliser des tests utilisateurs plus complets, c'est ce que nous avons appelé les démonstrations rapides. Lors de ces tests, toutes les fonctionnalités du logiciel étaient opérationnelles, ce qui n'était pas le cas lors des simulations rapides (par exemple, le changement d'université n'était pas possible car le logiciel ne contenait que des informations sur Namur).

7.2.5. Tests supplémentaires

Une étape à ne pas négliger lors du développement d'un logiciel pour tout public est certainement celle des tests utilisateurs. Si le respect des différents critères et règles ergonomiques est sans aucun doute le premier pas pour obtenir des interfaces valables, le

deuxième, nous semble-t-il, est de réaliser des tests pour confirmer ou infirmer les choix déjà opérés.

Avec la méthodologie que nous avons choisie, les tests se sont déroulés depuis la première étape. Il nous semblait en effet indispensable d'avoir un feed-back rapide sur la qualité et la compréhension de nos interfaces.

Nous devons également dire que nous n'avons pas effectué de réelles mesures statistiques lors de nos tests. Ceux-ci n'ont été effectués que dans l'optique d'améliorer Télémaque. Les remarques des testeurs ont donc été prises en compte directement.

a. Les testeurs

Nous devons malheureusement avouer que la population de nos testeurs est assez réduite puisque seulement une douzaine de personnes la compose. Mais à part deux personnes qui travaillaient avec nous, tous les autres étaient des étudiants susceptibles d'utiliser Télémaque. Les testeurs représentaient donc très bien la population réelle d'utilisateurs.

b. La procédure de test

Nos tests se sont effectués soit de manière individuelle, soit par deux testeurs. La procédure se déroulait en trois temps.

Lors de la première partie, nous demandions aux testeurs de réaliser quelques tâches précises telles que trouver des renseignements sur les logements pour l'université de Namur. Sans leur donner de conseils quant à la réalisation de cette tâche, nous les observions; nous notions les erreurs effectuées et leurs hésitations.

Pendant la deuxième partie des tests, nous demandions à quoi faisaient penser les différentes icônes présentes sur les boutons graphiques du menu graphique. Cet exercice nous a permis de valider la plupart d'entre elles.

La dernière partie était consacrée aux remarques ouvertes des utilisateurs. Mais il nous faut dire que celle-ci n'a pas été à la hauteur de nos espérances. Les étudiants semblaient en fait fort satisfaits du prototype que nous leur proposons. Nous pensons que ce manque de critique vient du fait qu'un tel logiciel n'est pas encore courant et que les testeurs n'avaient donc comme point de repère que la tâche traditionnelle, qui selon leurs propres mots était assez ennuyeuse.

Globalement, nous avons pu constater que les tests effectués par paire d'utilisateurs se déroulaient mieux. Une certaine émulation se produisait et les échanges verbaux entre les utilisateurs étaient souvent révélateurs de petits problèmes tels que des libellés ambigus.

A posteriori, nous nous sommes aperçus que cette procédure présentait des lacunes et qu'il aurait été préférable de suivre une démarche plus formelle.

Nous aurions dû, pour réaliser les tests, suivre plus rigide-ment la procédure conseillée par [LAU90].

1. Mise en place de l'observation. Ce premier point se compose de trois parties. Premièrement, il faut concevoir une tâche à faire réaliser par les testeurs. Cette tâche tournera autour de la partie du produit que l'on veut étudier. Deuxièmement, il faudra recruter les utilisateurs. La seule contrainte à respecter ici est d'essayer de trouver des personnes qui ont le même profil que les vrais utilisateurs. Le dernier point sera d'aménager une salle pour effectuer ces tests. Pour bien faire, cette pièce devra rassembler les conditions normales d'utilisation du logiciel.

Nous pouvons dire que pour cette première étape, nous n'avons pas respecté le dernier point. Il nous était en effet difficile de recréer les conditions normales d'utilisation de la borne, celle-ci se faisant dans le hall de l'IECS.

2. Description du but de l'observation. Pour ne pas stresser l'utilisateur, il faut bien lui expliquer que ce n'est pas lui que l'on va observer, mais le logiciel. Il faut lui faire comprendre qu'il fait partie de l'équipe de développement et qu'il n'est certainement pas un sujet d'observation.

Nous n'avons pas été aussi explicite lors de nos tests. Par contre, aucune remarque de notre part n'allait à l'encontre des testeurs.

3. Prévenir l'utilisateur qu'il peut partir quand il le veut. Dans l'idée de réaliser des tests non tronqués, il faut que le testeur soit dans de bonnes conditions de travail. Donc, si à un moment donné, il se sent mal à l'aise, celui-ci doit pouvoir quitter la séance de test immédiatement.

Nous n'avons pas non plus parlé de ceci aux testeurs. Comme ils venaient de leur plein gré aux séances de tests, nous ne voyions pas la nécessité de les prévenir du fait qu'ils pouvaient quitter la séance de tests quand ils le voulaient.

4. Description et démonstration du matériel. Il faut expliquer chaque pièce d'équipement présente dans la salle d'observation, aussi bien au niveau du logiciel à tester (clavier, écran, souris, etc.) qu'au niveau des appareils de "surveillance" utilisés (microphone, caméra, etc.).

Ici aussi, nous n'avons pas pris la peine d'expliquer à quoi servait tout le matériel. Nous signalions simplement que l'interactivité entre l'homme et la machine se faisait via un écran tactile.

5. Expliquer comment penser à voix haute. Il est très intéressant de savoir ce que pense le testeur lorsqu'il travaille sur le logiciel. Il faut donc lui expliquer qu'il doit penser à voix haute pendant toute la durée des opérations.

Nous n'avons jamais expliqué comment penser à haute voix. Nous demandions simplement aux testeurs de faire leurs remarques quand ils en avaient.

6. Expliquer que l'on ne pourra pas fournir d'aide. Le testeur devra également comprendre que pendant le test, vous ne pourrez pas lui donner d'explication directement. Il faudra fixer un temps minimum de réflexion (par exemple trois minutes) après lesquels l'examineur pourra également intervenir. Mais le testeur pourra noter ces questions pour qu'un débat puisse se faire après l'expérience.

Comme nous l'avons dit au point précédent, nous ne refusions pas le dialogue avec le testeur pendant les tests. Mais il est vrai que nous attendions un peu pour voir si le testeur pouvait se sortir lui même d'un mauvais pas ou non.

7. Décrire les tâches et introduire le sujet. Ceci est l'avant-dernière étape avant le commencement des tests. L'utilisateur devra recevoir les instructions concernant ce qu'il devra faire et on devra lui expliquer la fonction globale du produit.

Bien entendu, lors des tests, nous décrivions les tâches que les testeurs devraient réaliser.

8. Répondre aux éventuelles questions puis commencer les tests. Avant de commencer les tests, il faut demander à l'utilisateur s'il n'a pas quelques questions à poser. Si ce n'est pas le cas, on peut débiter.

Cette petite séance de questions réponses se faisait aussi lors des tests effectués sur Télémaque.

9. Conclusion de l'observation. A la fin de l'observation, on peut répondre aux questions de l'utilisateur. Après cela, on peut lui expliquer ce que l'on recherchait au travers de ces tests. Il est également intéressant de lui demander ces impressions en général.

Comme nous l'avons déjà mentionné, nous répondions aux questions pendant les tests. Mais nous demandions quand même l'impression générale après l'utilisation et les remarques éventuelles.

10. Utilisation des résultats. La tendance chez un observateur, lorsqu'une manipulation se passe mal, est de tout mettre sur le dos du testeur. Une telle attitude est évidemment stupide. Si un problème survient, il est important de prendre bien note de celui-ci pour comprendre pourquoi il s'est produit. De cette manière, il a des chances d'être résolu.

Bien entendu, nous respectons ce point. La découverte des faiblesses de nos interfaces étant le but principal de nos tests, nous étions très attentifs aux problèmes rencontrés par les testeurs.

Ce chapitre clôture la réalisation de la borne Télémaque. Nous y avons explicité le choix de notre plate-forme de développement et avons donné un sens pratique aux étapes de notre méthodologie. Dans le chapitre suivant, nous concluons ce travail en précisant quel était notre objectif, ce que nous avons fait pour l'atteindre et quels sont les développements futurs de Télémaque.

8. Conclusion

L'objectif de notre mémoire était la conception et la réalisation d'une borne interactive multimédia. Pour atteindre celui-ci, nous avons dû trouver une méthodologie adéquate. Nous avons donc passé en revue les méthodologies existantes (le modèle de la cascade, l'approche par prototypage, l'approche opérationnelle et l'approche mixte). Sur base de ces différentes méthodologies, nous avons alors présenté celle utilisée pour développer Télémaque. Nous voulions que celle-ci prenne fortement en compte les tests avec les utilisateurs. Pour rappel, cette méthodologie se décompose en quatre grandes étapes qui sont le prototypage non-opérationnel, les simulations rapides, les démonstrations rapides et les tests supplémentaires. Les deux points les plus développés dans ce travail étant le prototypage non-opérationnel et les simulations rapides.

Le premier de ces deux éléments a été réalisé grâce à un scénario. Nous en avons profité pour approfondir cette notion. Nous avons en fait étudié les rapports entre les scénarios cinématographique et publicitaire et le scénario informatique. Pour la partie concernant le scénario cinématographique, nous nous sommes essentiellement basés sur l'ouvrage de [CHI85]. Pour le scénario publicitaire, notre référence principale est [PEE86]. Cette étude a débouché sur une définition de scénario informatique et sur l'énumération de ses principales caractéristiques.

Le deuxième élément reprend essentiellement la création des interfaces et plus particulièrement des écrans de notre borne interactive. Pour développer ce point nous nous sommes inspirés de [VAN94]. De plus, nous avons identifié certaines règles ergonomiques grâce à l'expérience acquise tout au long du développement de Télémaque. Dans la partie traitant ce sujet, nous retrouvons quelques définitions avant de présenter les critères ergonomiques de design que nous avons trouvés dans [VAN94]. Nous les avons d'abord étudiés en théorie avant de les appliquer à la réalisation de Télémaque. En plus de cela, nous avons présenté les écrans de Télémaque sous différents points avant d'en proposer une évaluation.

Afin de réaliser ces écrans, nous avons également dû choisir un langage de développement. Nous avons, après une petite étude de marché sur les logiciels de présentation, les systèmes auteurs et les langages de programmation, arrêté notre choix sur le langage de programmation Visual Basic Pro.

Notre apport principal dans ce travail se situe à deux endroits: lors de l'étude du scénario informatique et dans le domaine de l'ergonomie. Le premier point présente un certain degré d'originalité: l'analogie entre le scénario cinématographique a permis d'identifier des caractéristiques valables pour le scénario informatique alors que le scénario publicitaire indique plutôt une marche à suivre pour l'élaboration d'un scénario informatique. Le deuxième point s'illustre par la synthèse des règles ergonomiques que nous avons reprises.

Pour terminer ce travail, nous pouvons citer les différentes extensions possibles concernant Télémaque. Dans un premier temps, on pourrait envisager d'étendre le nombre de bornes interactives multimédia sur le campus. Il faudrait alors relier celles-ci via un réseau pour permettre une gestion raisonnable de la maintenance. Dans un second temps, il faudrait connecter Télémaque et sa base de données aux différentes universités partenaires reprises dans Télémaque. De cette manière, ces universités auraient accès aux informations et elles pourraient mettre à jour elles-mêmes les informations les concernant. Le dernier projet serait tout à fait différent des deux premiers. Il s'agirait de mettre Télémaque, avec les informations ne concernant que l'IECS, sur CD-ROM. Celui-ci pourrait alors servir de support publicitaire pour l'école de commerce de Strasbourg.

9. Bibliographie

- [AGR86] Agresti, W.W., *New Paradigms for Software Development*, IEEE Computer Society Press / North-Holland, 1986.
- [BLA92a] Blattner, M., Multimedia and Multimodal User Interface Design, *ACM Conference on human factors in computing systems*, 1992.
- [BLA92b] Blattner, M. & Dannenberg R., *Multimedia Interface Design*, Addison-Wesley, 1992.
- [BOD89] Bodart, F., *Cours introductif aux interfaces homme - machine*, notes de cours provisoires du chapitre I, année académique 1989-1990.
- [CAR94] Carroll, J.M., Making Use a Design Representation, *Communications of the ACM*, vol. 37 n°12, pages 29-35, décembre 1994.
- [CHI85] Chion, M., *Ecrire un scénario*, Cahiers du cinéma / INA, 1985.
- [CRA94] Crampes, M., Eléments pour une modélisation de la communication multimédia, *Interface to real and virtual worlds*, Montpellier, 7-11 février 1994.
- [DIT93] Dittrich, S., *Visual Basic 3.0 pour Windows*, Micro Application, 1993.
- [EIS94] Eisenkolb, K. & Weickardt H., *Le grand livre de la vidéo sur PC*, Editions Micro Application, 1994
- [FIS94] Fisher, S., *Multimedia Authoring*, AP Professional, 1994.
- [GIM94] Gimenez, J-M & Daly S., Tous les secrets du multimédia, *Windows plus n°12*, septembre 1994.
- [GOF95] Goffinet, L., *Le multimédia et ses systèmes auteurs*, Institut d'informatique, FUNDP, Namur, 1995.
- [GOR95] Goray, E., *Les technologies du multimédia*, Namur (journée du multimédia), 25 janvier 1995.
- [GOU87] Gould, J., Boies S., Levy S., Richards J. & Schoonard J., The 1984 Olympic Message System: a Test of Behavioral Principles of System Design, *Communications of the ACM*, vol. 30 n°9, septembre 1987.
- [GRE93] Grebe, A.H., Burns M.J. & Weiss S.D., Interactive Multimedia Authoring Platforms, *ACM Conference on Human Factors in Computing Systems*, 1993.

- [HEM94] Hemery, A., Digitales attractions, *Multimédia solutions* n°21, juillet-août 1994.
- [HER94] Herellier, J-M, Vidéo: jouez la bonne carte, *Multimédia solutions* n°16, février 1994.
- [HSI94]) Hsia, P., Samuel J., Gao J., Kung D., Toyoshima Y. & Chen C., *Formal Approach to Scenario Analysis*, IEEE Software, mars 1994.
- [ISO91] ISO 9000-3, *Lignes directrices pour l'application de L'ISO 9001 au développement, à la mise à disposition et à la maintenance du logiciel*, 1991.
- [KAR83] Karnas, G. & Salengros P., L'ergonomie: adapter?, *La revue nouvelle*, vol. 3, 1983.
- [KAU89] Kaufman, L.D., Thebaut S.M. & Interrante M.F., *System Modeling for Scenario-Based Requirements Engineering*, Technical Report SERC-TR-33-F, Software Engineering Research Center, Computer and Information Sciences Department, University of Florida, septembre 1989.
- [KOE94] Koegel Buford, J., *Multimedia Systems*, Addison-Wesley, 1994.
- [LAU90] Laurel, B. (Ed.), *The Art of Human-Computer Interface Design*, Addison-Wesley, 1990.
- [LAU92] Laufer, R. & Scavetta D., *Texte, hypertexte et hypermédia*, Presses universitaires de France, 1992.
- [LOE92] Loeb, S., Architecting personalized delivery of multimedia information, *Communications of the ACM*, vol. 35 n°12, page 39, 1992.
- [MYL95] Mylopoulos, J., *Conceptual Modeling for Information Systems Engineering*, Chaire internationale, Namur, 1995.
- [NIL93] Nilsson, J., Khakhar D. & Sonesson B., Design of Living Book in Multimédia Medical Applications, *ICCC Conference on Multimedia Communications'93*, Banff, Alberta, Canada, 13-16 avril 1993.
- [NOR86] Norman, D.A. & Draper S.W., *User Centered System Design - New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum, Hillsdale, 1986.
- [OLI93] Olivetti Kioskit, 1993.
- [PAL94] Pallix, S., Comment faire du montage virtuel, *Multimedia solutions* n°17, mars 1994.

- [PEE86] Peeters, B., *Autour du scénario*, Editions de l'Université de Bruxelles, 1986.
- [PIC93] Pichault, F., *La borne interactive multimédia: une nouvelle technologie de distribution de l'information ?*, Université de Liège, 1993.
- [POT94] Potts, C., Takahashi K. & Anton A., Inquiry-Based Requirements Analysis, *IEEE Software*, pages 21-32, mars 1994.
- [RAU95] Rauterberg, M., *Four different measures to quantify three usability attributes: "feedback", "interactive directness" and "flexibility"*, in 2nd Eurographics Workshop on Design, Specification, Verification of interactive systems, DSV-IS'95.
- [RUB86] Rubens, P., Online Information, Traditional page design, And Reader Expectation, *IEEE Transactions on Professional Communication*, partie contractante-29, vol.4, 1986.
- [SHN87] Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, Reading (Massachusetts), 1987.
- [TAC95] Tachet, E., *La norme MIDI*, formation dispensée à La Walck (Strasbourg), janvier 1995.
- [VAA93] Vaananen, K., Interfaces to Hypermedia: Communicating the Structure and Interaction Possibilities to the Users, *Computer and Graphics*, vol. 17 n°3, 1993.
- [VAN92] Vanderdonckt, J., Sacre B. & Provot-Sacre I., *Une description orientée objet des objets interactifs abstraits utilisés en interface homme-machine*, rapport de recherche IHM/Ergo/10, 1992.
- [VAN94] Vanderdonckt, J., *Guide ergonomique des interfaces homme-machine*, Presses Universitaires de Namur, 1995.
- [VAN95] Vanderdonckt, J., *Génération automatique d'une interface homme-machine ergonomique pour une application de gestion hautement interactive*, à paraître, 1995.

10. Annexes

ANNEXE 1. CODE DE TÉLÉMAQUE

ANNEXE 2. SCÉNARIO DE TÉLÉMAQUE

ANNEXE 3. MANUEL D'EXPLOITATION DE LA BASE DE DONNÉES

ANNEXE 1. CODE DE TÉLÉMAQUE

Visual Basic offre une programmation événementielle et non procédurale. Il en découle que le code de Télémaque est difficilement lisible car il n'y a pas de hiérarchie. Dans d'autres langages tels que Cobol et Pascal, on peut identifier une procédure principale faisant appel à des routines. Dans Visual Basic, il n'y a ni procédure principale ni routines. Il n'est donc pas possible de présenter une architecture avec des modules.

Visual Basic comprend des contrôles et des événements. Les contrôles sont la fenêtre, le bouton, l'image, l'étiquette, le contrôle du multimédia, l'horloge... Les événements sont l'ouverture, la fermeture, l'activation, le click, le double-click...

A chaque événement est associé une procédure qui est exécutée lorsque l'événement se produit. A côté des ces procédures, on peut définir des modules et y insérer des variables et procédures globales à toutes les fenêtres.

Le code qui est présenté dans les pages qui suivent a été généré par Visual Basic et reprend les procédures de chaque événement de chaque objet ainsi que les modules. Ces procédures et modules mis les uns à la suite des autres ne facilitent pas la lecture du code.

Le code est structuré de la façon suivante:

- *.FRM est le nom d'une fenêtre et - x, le numéro de page (relatif à une fenêtre);
- après le nom de la fenêtre sont déclarées les variables (Dim) et constantes (Const) de la fenêtre;
- lorsque le mot Sub apparaît, cela signifie qu'on commence une procédure qui sera clôturée par End Sub;
- à côté de Sub, on trouve X_Y: X correspond à un contrôle et Y, à un événement;
- *.BAS représente le nom d'un module.

AUBIBLIO.FRM - 1

```
Dim compteur As Integer
Const nbre = 4
```

```
Sub CmdPl_Click ()
If boolaide = False Then
tabprecform(15) = ecrancour
Plans.Show
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabpl(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdPrint_Click ()
Dim strtitre As String
Dim nbrp As Integer
```

```
If boolaide = False Then
cmdprint.Enabled = False
timer1.Enabled = True
If improk() = True Then
    On Error GoTo errhandler
    printer.ScaleMode = 7
    compteur = 0
    If (UBound(tabinitbiblionom, 1) Mod nbriems = 0) Then
        nbrp = UBound(tabinitbiblionom, 1) / nbriems
    Else
        nbrp = (UBound(tabinitbiblionom, 1) \ nbriems) + 1
    End If
    printer.FontName = "Times New Roman"
    printer.FontSize = 30
    printer.FontBold = True
    strtitre = req_ville + " : " + sstitre(1)
    printer.CurrentX = (printer.Width / 567 - Len(strtitre) * .5) / 2
    printer.Print (strtitre)
    printer.ScaleLeft = -2
    printer.Print
    printer.FontSize = 14
    If REQ_INFO = "Dates à retenir" Then
        While compteur < UBound(tabinitbiblionom, 1)
            compteur = compteur + 1
            printer.FontBold = True
            printer.Print tabinitbiblionom(compteur, 1)
            printer.FontBold = False
            printer.Print tabinitbibliodated(compteur) + " - " + tabinitbibliodatef(compteur)
            printer.FontItalic = True
            printer.Print tabinitbibliorem(compteur, 1)
            printer.FontItalic = False
            printer.Print
            If (compteur Mod nbriems = 0) And (compteur <> UBound(tabinitbiblionom, 1)) Then
                printer.CurrentX = 16
                printer.CurrentY = 26
                printer.Print (printer.Page + "/" + Str(nbrp))
                printer.NewPage
                printer.ScaleLeft = 0
                printer.FontSize = 30
                printer.FontBold = True
                printer.CurrentX = (printer.Width / 567 - Len(strtitre) * .5) / 2
                printer.Print (strtitre)
                printer.ScaleLeft = -2
                printer.Print
                printer.FontSize = 14
            End If
        Wend
        printer.CurrentX = 16
        printer.CurrentY = 26
        printer.Print (printer.Page + "/" + Str(nbrp))
```

```

Else
    While compteur < UBound(tabinitbiblionom, 1)
        compteur = compteur + 1
        printer.FontBold = True
        printer.Print tabinitbiblionom(compteur, 1)
        printer.FontBold = False
        printer.Print tabinitbibliorens(compteur)
        printer.FontItalic = True
        printer.Print tabinitbibliorem(compteur, 1)
        printer.FontItalic = False
        printer.Print
        If (compteur Mod nbrites = 0) And (compteur <> UBound(tabinitbiblionom, 1)) Then
            printer.CurrentX = 16
            printer.CurrentY = 26
            printer.Print (printer.Page + "/" + Str(nbrp))
            printer.NewPage
            printer.ScaleLeft = 0
            printer.FontName = "Times New Roman"
            printer.FontSize = 30
            printer.FontBold = True
            printer.CurrentX = (printer.Width / 567 - Len(strtitre) * .5) / 2
            printer.Print (strtitre)
            printer.ScaleLeft = -2
            printer.Print
            printer.FontSize = 14
            printer.FontBold = False
        End If
    Wend
    printer.CurrentX = 16
    printer.CurrentY = 26
    printer.Print (printer.Page + "/" + Str(nbrp))
End If
printer.EndDoc
End If
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabprnt(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
fin: Exit Sub
errhandler:
Resume fin
End Sub

Sub Form_Activate ()
If nonfirst = True Then
    ecrancour = 23
    lbltitre = req_ville
    InitBiblio
    compteur = 0
    cp = 1
    initbibliobis
    nbrimprloc = 0
    prochisto
End If
End Sub

Sub Form_Deactivate ()
compteur = 0
While compteur < nbre
    lblnom(compteur).Visible = False
    lbladr(compteur).Visible = False
    lbldescr(compteur).Visible = False
    compteur = compteur + 1
Wend
suiv.Visible = False
prec.Visible = False
End Sub

```



```
Sub Form_Unload (cancel As Integer)
cancel = True
End Sub
```

```
Sub Picture1_Click ()
guignols
End Sub
```

```
Sub prec_Click ()
If boolaide = False Then
    suiv.Visible = True
    compteur = 0
    While compteur < nbre
        lblnom(compteur).Visible = False
        lbladr(compteur).Visible = False
        lbldescr(compteur).Visible = False
        compteur = compteur + 1
    Wend
    compteur = 0
    cp = cp - nbre
    If cp <= 1 Then 'LBound(tabinitbiblio) Then
        prec.Visible = False
    End If
    While compteur < nbre
        lblnom(compteur).Caption = tabinitbiblionom(cp, 1)
        If REQ_INFO = "Dates à retenir" Then
            If unedate(cp) = False Then
                lbladr(compteur).Caption = tabinitbibliodated(cp) + " - " + tabinitbibliodatef(cp)
            Else
                lbladr(compteur).Caption = tabinitbibliodated(cp)
            End If
        Else
            lbladr(compteur).Caption = tabinitbibliorens(cp)
        End If
        lbldescr(compteur).Caption = tabinitbibliorem(cp, 1)

        lblnom(compteur).Visible = True
        lbladr(compteur).Visible = True
        lbldescr(compteur).Visible = True
        cp = cp + 1
        compteur = compteur + 1
    Wend
    cp = cp - nbre
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabhlp2(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub suiv_Click ()
If boolaide = False Then
    prec.Visible = True
    compteur = 0
    While compteur < nbre
        lblnom(compteur).Visible = False
        lbladr(compteur).Visible = False
        lbldescr(compteur).Visible = False
        compteur = compteur + 1
    Wend
    compteur = 0
    cp = cp + nbre
    While (cp <= nbrinfos) And (compteur < nbre)
        lblnom(compteur).Caption = tabinitbiblionom(cp, 1)
        If REQ_INFO = "Dates à retenir" Then
            If unedate(cp) = False Then
```

```
        lbladr(compteur).Caption = tabinitbibliodated(cp) + " - " + tabinitbibliodatef(cp)
    Else
        lbladr(compteur).Caption = tabinitbibliodated(cp)
    End If

    Else
        lbladr(compteur).Caption = tabinitbibliorens(cp)
    End If
    lbldescr(compteur).Caption = tabinitbibliorem(cp, 1)

    lblnom(compteur).Visible = True
    lbladr(compteur).Visible = True
    lbldescr(compteur).Visible = True
    cp = cp + 1
    compteur = compteur + 1
Wend
If cp > nbrinfos Then
    suiv.Visible = False
End If
cp = cp - compteur
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabhlp2(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
End Sub

Sub Timer1_Timer ()
cmdprint.Enabled = True
End Sub
```



```
'Indique si un item de la liste déroulante a été sélectionné
Dim choixfait As Integer
'Indique d'où vient l'utilisateur
Dim douvinsje As Integer
```

```
Sub scroll_Change ()
```

```
End Sub
```

```
Sub ascen_Change ()
```

```
If boolaide = False Then
```

```
For m% = 0 To 9
```

```
    label1(m%).BackColor = &H80000005
```

```
    label1(m%).ForeColor = &H80000008
```

```
Next m%
```

```
For p% = 0 To 9
```

```
    If tabconcat(ascen.Value + p% + 1).indice = 9999 Then
```

```
        label1(p%).Alignment = 2
```

```
        label1(p%).FontSize = 15
```

```
        label1(p%).FontItalic = True
```

```
        label1(p%).Caption = tabenspays(tabconcat(ascen.Value + p% + 1).pay, 1)
```

```
        label1(p%).Tag = "9999"
```

```
        label1(p%).ForeColor = &HB14B07
```

```
    Else
```

```
        label1(p%).FontSize = 12
```

```
        label1(p%).FontItalic = False
```

```
        label1(p%).Alignment = 0
```

```
        label1(p%).Caption = tabconcat(ascen.Value + p% + 1).viluniv
```

```
        label1(p%).Tag = Str(tabconcat(ascen.Value + p% + 1).indice)
```

```
    End If
```

```
Next p%
```

```
    choixfait = False
```

```
    cmdok.Visible = False
```

```
ElseIf boolfin = True Then
```

```
boolfin = False
```

```
MDI1.MM1.FileName = tabhlp2(1)
```

```
MDI1.MM1.Command = "open"
```

```
MDI1.MM1.Command = "play"
```

```
End If
```

```
End Sub
```

```
Sub cmdok_Click ()
```

```
If boolaide = False Then
```

```
If choixfait = True Then
```

```
    prechisto = False
```

```
    req_pays = tabchgtuniv(choixuniv).pa
```

```
    req_ville = tabchgtuniv(choixuniv).vi
```

```
    req_univ = tabchgtuniv(choixuniv).un
```

```
MDI1!Data1.RecordSource = "select NOM_F from FACULTE WHERE (NOM_P = '"' + req_pays + '"' ) and (
```

```
M_V= '"' + req_ville + '"' ) and (NOM_U= '"' + req_univ + '"' )"
```

```
MDI1!Data1.Refresh
```

```
If MDI1!Data1.Recordset.EOF = False Then
```

```
    req_fac = MDI1!Data1.Recordset("NOM_F")
```

```
End If
```

```
'Pour réinitialiser les titres de certains écrans
```

```
'avec le nom du pays ou de la ville
```

```
pays = req_pays
```

```
If douvinsje = False Then
```

```
    tabidform(tabprecform(ecrancour)).Show
```

```
Else
```

```
MDI1!Data1.RecordSource = "select NOM_U from UNIVERSITE WHERE (NOM_P = '"' + req_pays + '"'
```

```
and (NOM_V= '"' + req_ville + '"' ) "
```

```
MDI1!Data1.Refresh
```

```
MDI1!Data1.Recordset.MoveLast
```

```
If MDI1!Data1.Recordset.RecordCount > 1 Then
```

```
    tabprecform(21) = 26
```

```
Else
```

```
    tabprecform(21) = 20
```

```
End If
```

```
menuville.Show
```

```

    End If
End If
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

Sub Command3D1 Click ()
If boolaide = False Then
    If ascen.Value <> 0 Then
        ascen.Value = ascen.Value - 1
    End If
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

Sub Command3D2 Click ()
If boolaide = False Then
    If ascen.Value < ascen.Max Then
        ascen.Value = ascen.Value + 1
    End If
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

Sub Form_Activate ()
If nonfirst = True Then
    Dim nbrtabconcat As Integer
    Dim dejavu As Integer
    Dim tailletabchgtuniv As Integer

    initchgtuniv
    tabprecform(25) = ecrancour
    prechisto = True

    'tester les 4 écrancour, si l'utilisateur vient avant
    'l'écran Introville, alors on l'enmène directement sur
    'cet écran.
    Select Case ecrancour
        Case 2, 19, 20, 21, 24, 26
            douvinsje = True
        Case Else
            douvinsje = False
    End Select

    ecrancour = 25

    'Remplissage du dynaset ds contenant tous les pays présents
    'dans la Base de données (redondant mais utile si on vient
    'de l'écran suivant)
    MDI1!Data1.RecordSource = "select NOM_P from PAYS"
    MDI1!Data1.Refresh
    Set ds = MDI1!Data1.Recordset

    MDI1!Data1.RecordSource = "select NOM_P, NOM_V, NOM_U from UNIVERSITE order by NOM_P, NOM_V, I
M_U"
    MDI1!Data1.Refresh
    Set ds_concat = MDI1!Data1.Recordset
    ds_concat.MoveLast
    ReDim tabchgtuniv(ds_concat.RecordCount)
    ds_concat.MoveFirst
    'Remplissage du tableau tabchgtuniv
    i% = 1
    While Not ds_concat.EOF

```



```

    If ds_concat("NOM_P") <> "#NULL#" Then tabchgtuniv(i%).pa = ds_concat("NOM_P")
    If ds_concat("NOM_V") <> "#NULL#" Then tabchgtuniv(i%).vi = ds_concat("NOM_V")
    If ds_concat("NOM_U") <> "#NULL#" Then tabchgtuniv(i%).un = ds_concat("NOM_U")
    ds_concat.MoveNext
    i% = i% + 1
Wend
tailletabchgtuniv = i% - 1
'Remplissage du tableau Tabconcat (qui sera affiché dans la liste défilante)
'initialisation des variables
dejavu = False
i% = 1
j% = 1
ReDim tabconcat(j%)
pointeur% = 1
While (i% <= tailletabchgtuniv) And (pointeur% <= UBound(tabpaysdans))
    If tabenspays(tabpaysdans(pointeur%), 1) = tabchgtuniv(i%).pa Then
        If dejavu = False Then
            tabconcat(j%).pay = tabpaysdans(pointeur%)
            tabconcat(j%).viluniv = ""
            tabconcat(j%).indice = 9999
            dejavu = True
        Else
            tabconcat(j%).pay = tabpaysdans(pointeur%)
            tabconcat(j%).viluniv = tabchgtuniv(i%).vi + " : " + tabchgtuniv(i%).un
            tabconcat(j%).indice = i%
            i% = i% + 1
        End If
        j% = j% + 1
        ReDim Preserve tabconcat(j%)
    Else
        pointeur% = pointeur% + 1
        dejavu = False
    End If
Wend

'Assignation de la valeur maximum pour l'ascenceur
ascen.Max = UBound(tabconcat) - 11
ascen.Visible = True
command3d1.Visible = True
command3d2.Visible = True
If UBound(tabconcat) > 11 Then
    ascen.Enabled = True
    command3d1.Enabled = True
    command3d2.Enabled = True
End If

For i% = 0 To 9
    labell(i%).Caption = ""
Next
'Remplissage de la liste défilante
i% = 1
While (i% <= 10) And (i% <= UBound(tabconcat) - 1)
    If tabconcat(i%).indice = 9999 Then
        labell(i% - 1).FontSize = 15
        labell(i% - 1).FontItalic = True
        labell(i% - 1).ForeColor = &HB14B07
        labell(i% - 1).Alignment = 2
        labell(i% - 1).Caption = tabenspays(tabconcat(i%).pay, 1)
        labell(i% - 1).Tag = "9999"
    Else
        labell(i% - 1).FontSize = 12
        labell(i% - 1).FontItalic = False
        labell(i% - 1).Caption = tabconcat(i%).viluniv
        labell(i% - 1).Tag = Str(tabconcat(i%).indice)
        labell(i% - 1).Alignment = 0
    End If
    labell(i% - 1).Enabled = True
    i% = i% + 1
Wend
For i% = 0 To 9
    labell(i%).Visible = True
Next
End If
End Sub

```

```

Sub Form_Deactivate ()
For t% = 0 To 9
    labell(t%).BackColor = &H80000005
    labell(t%).ForeColor = &H80000008
    labell(t%).Caption = ""
    labell(t%).Visible = False
    labell(t%).Enabled = False
    ascen.Visible = False
    ascen.Enabled = False
    command3d1.Visible = False
    command3d1.Enabled = False
    command3d2.Visible = False
    command3d2.Enabled = False
    ascen.Value = 0
Next t%
cmdok.Visible = False
End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Labell_Click (index As Integer)
For m% = 0 To 9
    labell(m%).BackColor = &H80000005
    If labell(m%).ForeColor = &HFFFFFF Then
        labell(m%).ForeColor = &H0&
    End If
Next m%

If labell(index).Tag <> "9999" Then
    labell(index).BackColor = &H0&
    labell(index).ForeColor = &HFFFFFF
    choixuniv = Val(labell(index).Tag)
    choixfait = True
    cmdok.Visible = True
Else
    choixfait = False
    cmdok.Visible = False
End If
End Sub

Sub Picture1_Click ()
guignols
End Sub

```



```

Sub Cmdstop_Click ()
Dim ij As Integer

ij = mciExecute("stop ca")
ij = mciExecute("close ca")
fentv.Visible = False
fenpourtiv.Visible = False
MenuIECS.Show
End Sub

Sub Form_Activate ()
If nonfirst = True Then
    Dim ij As Integer
    Dim inst As String

    EcranCour = 11
    lbltitre.Caption = req_ville
    InitCondAdm
    MDI1!Data1.RecordSource = "select LOGO_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "' and
NOM_V='" + req_ville + "' and NOM_U='" + req_univ + "'"
    MDI1!Data1.Refresh
    Set ds_unif = MDI1!Data1.Recordset
    If ds_unif("LOGO_U") <> "#NULL#" Then
        On Error GoTo errorhandler
        picture2.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\UNIV\" + ds_unif("LOGO_U") + ".bmp")
        picture2.Visible = True
    End If
un:
    MDI1.Picture1.Visible = False
    fentv.Visible = True
    DoEvents
    fenpourtiv.Visible = True
    inst = "open " + chem + "TELEMAQU\France\Strasbou\univ\condadm.avi alias ca"
    ij = mciExecute(inst)
    inst = "window ca" + " handle " + Str(fenpourtiv.hWnd)
    j = mciExecute(inst)
    ij = mciExecute("play ca")
End If
Exit Sub
errorhandler:
Resume un:
End Sub

Sub Form_Deactivate ()
prechisto = True
picture2.Visible = False
End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Picture1_Click ()
guignols
End Sub

```

COURS.FRM - 1

```
Dim k As Integer
Const nbrlbl = 3
Dim compteur As Integer
Dim ds_coursimpr As dynaset
Dim nbrtotcours As Integer
```

```
Sub Anigroupe_Click (Index As Integer)
Dim sql As String
```

```
If boolaide = False Then
```

```
groupe = False
```

```
'Les boutons pour les groupes sont mis à invisible
cubis = Index
```

```
For i% = 1 To 9
    anigroupe(i%).visible = False
Next i%
cmdprintcrs.visible = False
```

```
'Requête pour les cours qui appartiennent au groupe sélectionné
```

```
sql = " Select INTITULEFRA, INTITULEANG, INTITULEALL, INTITULEITA, INTITULEESP, "
sql = sql + "PROF, [NBRE_HEURES] from COURS WHERE ANNEE='" + Right$(Str(cu), 1)
sql = sql + "' and NOM_P = '" + req_pays + "' and NOM_V='" + req_ville + "' and NOM_U='"
sql = sql + req_univ + "' and NOM_F='" + req_fac + "' and GROUPEFRA='" + tabtempgrp(cubis) + "
```

```
MDI1!Data1.RecordSource = sql
MDI1!Data1.Refresh
Set ds_cours = MDI1!Data1.Recordset
If ds_cours.EOF = False Then
```

```
    ds_cours.MoveLast
    bornesup = ds_cours.RecordCount
    ds_cours.MoveFirst
    'Le reste de l'écran apparait
    If bornesup > 3 Then
        scroll.Enabled = True
        command3d1.Enabled = True
        command3d2.Enabled = True
    End If
```

```
For i% = 1 To 3
    lblintit(i%).visible = True
    lblintit(i%).Caption = ""
    lblprof(i%).visible = True
    lblprof(i%).Caption = ""
    lblheures(i%).visible = True
    lblheures(i%).Caption = ""
Next i%
```

```
line1.visible = True
line2.visible = True
line3.visible = True
line4.visible = True
line5.visible = True
scroll.visible = True
command3d1.visible = True
command3d2.visible = True
ReDim tabintit(bornesup, nbl)
ReDim tabp(bornesup)
ReDim tabh(bornesup)
i% = 1
```

```
While Not (ds_cours.EOF)
```

```
    If ds_cours("INTITULEFRA") <> "#NULL#" Then tabintit(i%, 1) = " " + ds_cours("INTITUL
RA")
    If ds_cours("INTITULEANG") <> "#NULL#" Then tabintit(i%, 2) = " " + ds_cours("INTITUL
NG")
    If ds_cours("INTITULEALL") <> "#NULL#" Then tabintit(i%, 3) = " " + ds_cours("INTITUL
LL")
    If ds_cours("INTITULEITA") <> "#NULL#" Then tabintit(i%, 4) = " " + ds_cours("INTITUL
TA")
    If ds_cours("INTITULEESP") <> "#NULL#" Then tabintit(i%, 5) = " " + ds_cours("INTITUL
SP")
    If ds_cours("PROF") <> "#NULL#" Then tabp(i%) = " " + ds_cours("PROF")
    If ds_cours("NBRE_HEURES") <> "#NULL#" Then tabh(i%) = " " + ds_cours("NBRE_HEURES")
    "H"
    i% = i% + 1
    ds_cours.MoveNext
```



```

Wend

'Initialisation de la liste défilante
lstde% = 1
While (lstde% <= 3) And (lstde% <= bornesup)
    lblintit(lstde%).Caption = tabintit(lstde%, 1)
    lblprof(lstde%) = tabp(lstde%)
    lblheures(lstde%) = tabh(lstde%)
    lstde% = lstde% + 1
Wend

'Initialisation du scroll
scroll.Max = bornesup - 3
Else
    afficher
    Form_Activate
End If
ElseIf boolfin = True Then
    boolfin = False
    MDI1.MM1.FileName = tabhlp2(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
End Sub

Sub Cmdprintcrs_Click ()
    Dim strtitre As String
    Dim nbrp As Integer

    If boolaide = False Then
        cmdprintcrs.Enabled = False
        timer1.Enabled = True
    End If
    If improk() = True Then
        On Error GoTo errhandler
        printer.ScaleMode = 7
        compteur = 0
        If nbrtotcours Mod nbritems = 0 Then
            nbrp = nbrtotcours / nbritems
        Else
            nbrp = nbrtotcours \ nbritems + 1
        End If
        printer.FontName = "Times New Roman"
        printer.FontSize = 30
        printer.FontBold = True
        strtitre = req fac + " : " + tabannee(cu, 1)
        printer.CurrentX = (printer.Width / 567 - Len(strtitre) * .5) / 2
        printer.Print (strtitre)
        printer.ScaleLeft = -2
        printer.Print
        printer.FontSize = 14
        ds_coursimpr.MoveFirst
        While compteur < nbrtotcours
            compteur = compteur + 1
            printer.FontBold = True
            Select Case 1
                Case 1
                    printer.Print ds_coursimpr("INTITULEFRA")
                Case 2
                    printer.Print ds_coursimpr("INTITULEANG")
                Case 3
                    printer.Print ds_coursimpr("INTITULEALL")
                Case 4
                    printer.Print ds_coursimpr("INTITULEITA")
                Case 5
                    printer.Print ds_coursimpr("INTITULEESP")
            End Select
            printer.FontBold = False
            printer.Print ds_coursimpr("PROF")
            printer.FontItalic = True
            printer.Print (ds_coursimpr("[NBRE_HEURES]") + "H")
            printer.FontItalic = False
            printer.Print
            If (compteur Mod nbritems = 0) And (compteur <> nbrtotcours) Then
                printer.CurrentX = 16
                printer.CurrentY = 26
            End If
        End While
    End If

```

```

        printer.Print (printer.Page + "/" + Str(nbrp))
        printer.NewPage
        printer.ScaleLeft = 0
        printer.FontSize = 30
        printer.FontBold = True
        printer.CurrentX = (printer.Width / 567 - Len(strtitre) * .5) / 2
        printer.Print (strtitre)
        printer.ScaleLeft = -2
        printer.Print
        printer.FontSize = 14
    End If
    ds_coursimpr.MoveNext
Wend
printer.CurrentX = 16
printer.CurrentY = 26
printer.Print (printer.Page + "/" + Str(nbrp))
printer.EndDoc
End If
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabprnt(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
fin: Exit Sub
errhandler:
Resume fin
End Sub

Sub Command3D1_Click ()
    If boolaide = False Then
        If scroll.Value > 0 Then
            scroll.Value = scroll.Value - 1
        End If
    ElseIf boolfin = True Then
        boolfin = False
        MDI1.MM1.FileName = tabhlp2(1)
        MDI1.MM1.Command = "open"
        MDI1.MM1.Command = "play"
    End If
End Sub

Sub Command3D2_Click ()
    If boolaide = False Then
        If scroll.Value < bornesup - 3 Then
            scroll.Value = scroll.Value + 1
        End If
    ElseIf boolfin = True Then
        boolfin = False
        MDI1.MM1.FileName = tabhlp2(1)
        MDI1.MM1.Command = "open"
        MDI1.MM1.Command = "play"
    End If
End Sub

Sub Form_Activate ()
    If nonfirst = True Then
        Dim ds_grp As dynaset
        Dim ds_grptemp As dynaset
        Dim sql As String

        cours.lblsstitre.Caption = tabannee(cu, 1)
        'gestion de l'impression
        sql = "select * from COURS WHERE NOM_P = '" + req_pays
        sql = sql + "' and NOM_V=''" + req_ville + "' and NOM_U=''" + req_univ
        sql = sql + "' and NOM_F=''" + req_fac + "' and ANNEE=''" + Right$(Str(cu), 1)
        sql = sql + "' order by INTITULEFRA"
        MDI1!Data1.RecordSource = sql
        MDI1!Data1.Refresh
        Set ds_coursimpr = MDI1!Data1.Recordset
    End If
End Sub

```



```

If ds_coursimpr.EOF = False Then
    ds_coursimpr.MoveLast
    nbrtotcours = ds_coursimpr.RecordCount
    ds_coursimpr.MoveFirst
Else
    nbrtotcours = 0
End If
'
groupe = True
For i% = 1 To 5
    cursus!CmdAnnée(i%).visible = False
Next i%
lbttitre.Caption = req_ville

'Accès bd pour prendre les groupes
MDI1!Data1.RecordSource = "select distinct GROUPEFRA from COURS WHERE NOM_P = '" + req_pays +
' and NOM V=' + req_ville + "' and NOM U=' + req_univ + "' and NOM_F=' + req_fac + "' and ANN
='" + Right$(Str(cu), 1) + "' order by GROUPEFRA"
MDI1!Data1.Refresh
Set ds_grptemp = MDI1!Data1.Recordset
If ds_grptemp.EOF = False Then
    cmdprintcrs.visible = True
    ds_grptemp.MoveLast
    'Test supplémentaire pour éviter le plantage s'il y a plus de 9 groupes
    'dans une même année
    If ds_grptemp.RecordCount <= 9 Then
        ReDim tabgroupe(ds_grptemp.RecordCount, nbl)
    Else
        ReDim tabgroupe(9, nbl)
    End If
    ds_grptemp.MoveFirst

    MDI1!Data1.RecordSource = "select distinct GROUPEFRA,GROUPEANG, GROUPEALL, GROUPEITA, GRO
EESP from COURS WHERE GROUPEFRA <> '#NULL#' and NOM_P = '" + req_pays + "' and NOM V=' + req_v
le + "' and NOM U=' + req_univ + "' and NOM_F=' + req_fac + "' and ANNEE=' + Right$(Str(cu),
+ "' order by GROUPEFRA"
    MDI1!Data1.Refresh
    Set ds_grp = MDI1!Data1.Recordset
    i% = 1

    'Traitement du premier élément
    If ds_grp("groupefra") <> "#NULL#" Then tabgroupe(i%, 1) = ds_grp("groupefra")
    If ds_grp("groupeang") <> "#NULL#" Then
        tabgroupe(i%, 2) = ds_grp("groupeang")
    Else
        tabgroupe(i%, 2) = ds_grp("groupefra")
    End If
    If ds_grp("groupeall") <> "#NULL#" Then
        tabgroupe(i%, 3) = ds_grp("groupeall")
    Else
        tabgroupe(i%, 3) = ds_grp("groupefra")
    End If
    If ds_grp("groupeita") <> "#NULL#" Then
        tabgroupe(i%, 4) = ds_grp("groupeita")
    Else
        tabgroupe(i%, 4) = ds_grp("groupefra")
    End If
    If ds_grp("groupeesp") <> "#NULL#" Then
        tabgroupe(i%, 5) = ds_grp("groupeesp")
    Else
        tabgroupe(i%, 5) = ds_grp("groupefra")
    End If
    i% = i% + 1
    ds_grp.MoveNext
    While Not (ds_grp.EOF) And (i% <= ds_grptemp.RecordCount)
        If ds_grptemp("GROUPEFRA") = ds_grp("GROUPEFRA") Then
            If tabgroupe(i% - 1, 1) <> ds_grp("GROUPEFRA") Then
                If ds_grp("groupefra") <> "#NULL#" Then tabgroupe(i%, 1) = ds_grp("groupefra")
                If ds_grp("groupeang") <> "#NULL#" Then
                    tabgroupe(i%, 2) = ds_grp("groupeang")
                Else
                    tabgroupe(i%, 2) = ds_grp("groupefra")
                End If
                If ds_grp("groupeall") <> "#NULL#" Then
                    tabgroupe(i%, 3) = ds_grp("groupeall")

```

```

Else
    tabgroupe(i%, 3) = ds_grp("groupefra")
End If
If ds_grp("groupeita") <> "#NULL#" Then
    tabgroupe(i%, 4) = ds_grp("groupeita")
Else
    tabgroupe(i%, 4) = ds_grp("groupefra")
End If
If ds_grp("groupeesp") <> "#NULL#" Then
    tabgroupe(i%, 5) = ds_grp("groupeesp")
Else
    tabgroupe(i%, 5) = ds_grp("groupefra")
End If
i% = i% + 1
End If
ds_grp.MoveNext
Else
    ds_grptemp.MoveNext
End If
Wend

If i% <> 0 Then
    nbrgroupe = i% - 1
End If

'fin de deactivate cursus

ecrancour = 13
initcours
'Accès bd pour afficher le logo universitaire
MDI1!Data1.RecordSource = "select LOGO_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "
and NOM_V=''" + req_ville + "'" and NOM_U=''" + req_univ + "'"
MDI1!Data1.Refresh
Set ds_unif = MDI1!Data1.Recordset
If ds_unif("LOGO U") <> "#NULL#" Then
    On Error GoTo errorhandler
    picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(
q_ville, 8) + "\" + Left$(req_univ, 8) + "\" + Left$(req_pays, 8) + "\" + Left$(req_ville, 8) + "\" + Left$(req_univ, 8) + ".bmp")
    picture1.Visible = True
End If
un:
Select Case nbrgroupe
Case 1
    anigroupe(5).Visible = True
    anigroupe(5).Caption = tabgroupe(1, 1)
    tabtempgrp(5) = tabgroupe(1, 1)

Case 2
    anigroupe(2).Caption = tabgroupe(1, 1)
    anigroupe(2).Visible = True
    anigroupe(5).Caption = tabgroupe(2, 1)
    anigroupe(5).Visible = True
    tabtempgrp(2) = tabgroupe(1, 1)
    tabtempgrp(5) = tabgroupe(2, 1)

Case 3
    anigroupe(4).Caption = tabgroupe(1, 1)
    anigroupe(4).Visible = True
    anigroupe(5).Caption = tabgroupe(2, 1)
    anigroupe(5).Visible = True
    anigroupe(6).Caption = tabgroupe(3, 1)
    anigroupe(6).Visible = True
    tabtempgrp(4) = tabgroupe(1, 1)
    tabtempgrp(5) = tabgroupe(2, 1)
    tabtempgrp(6) = tabgroupe(3, 1)

Case 4
    anigroupe(2).Caption = tabgroupe(1, 1)
    anigroupe(2).Visible = True
    anigroupe(4).Caption = tabgroupe(2, 1)
    anigroupe(4).Visible = True
    anigroupe(6).Caption = tabgroupe(3, 1)
    anigroupe(6).Visible = True
    anigroupe(8).Caption = tabgroupe(4, 1)
    anigroupe(8).Visible = True

```



```

tabtempgrp(2) = tabgroupe(1, 1)
tabtempgrp(4) = tabgroupe(2, 1)
tabtempgrp(6) = tabgroupe(3, 1)
tabtempgrp(8) = tabgroupe(4, 1)

```

Case 5

```

anigroupe(1).Caption = tabgroupe(1, 1)
anigroupe(1).visible = True
anigroupe(3).Caption = tabgroupe(2, 1)
anigroupe(3).visible = True
anigroupe(5).Caption = tabgroupe(3, 1)
anigroupe(5).visible = True
anigroupe(7).Caption = tabgroupe(4, 1)
anigroupe(7).visible = True
anigroupe(9).Caption = tabgroupe(5, 1)
anigroupe(9).visible = True
tabtempgrp(1) = tabgroupe(1, 1)
tabtempgrp(3) = tabgroupe(2, 1)
tabtempgrp(5) = tabgroupe(3, 1)
tabtempgrp(7) = tabgroupe(4, 1)
tabtempgrp(9) = tabgroupe(5, 1)

```

Case 6

```

For i% = 1 To nbrgroupe
    anigroupe(i%).Caption = tabgroupe(i%, 1)
    anigroupe(i%).visible = True
    tabtempgrp(i%) = tabgroupe(i%, 1)
Next i%

```

Case 7

```

anigroupe(1).Caption = tabgroupe(1, 1)
anigroupe(1).visible = True
anigroupe(2).Caption = tabgroupe(2, 1)
anigroupe(2).visible = True
anigroupe(3).Caption = tabgroupe(3, 1)
anigroupe(3).visible = True
anigroupe(5).Caption = tabgroupe(4, 1)
anigroupe(5).visible = True
anigroupe(7).Caption = tabgroupe(5, 1)
anigroupe(7).visible = True
anigroupe(8).Caption = tabgroupe(6, 1)
anigroupe(8).visible = True
anigroupe(9).Caption = tabgroupe(7, 1)
anigroupe(9).visible = True
tabtempgrp(1) = tabgroupe(1, 1)
tabtempgrp(2) = tabgroupe(2, 1)
tabtempgrp(6) = tabgroupe(3, 1)
tabtempgrp(5) = tabgroupe(4, 1)
tabtempgrp(7) = tabgroupe(5, 1)
tabtempgrp(8) = tabgroupe(6, 1)
tabtempgrp(9) = tabgroupe(7, 1)

```

Case 8

```

anigroupe(1).Caption = tabgroupe(1, 1)
anigroupe(1).visible = True
anigroupe(2).Caption = tabgroupe(2, 1)
anigroupe(2).visible = True
anigroupe(3).Caption = tabgroupe(3, 1)
anigroupe(3).visible = True
anigroupe(4).Caption = tabgroupe(4, 1)
anigroupe(4).visible = True
anigroupe(6).Caption = tabgroupe(5, 1)
anigroupe(6).visible = True
anigroupe(7).Caption = tabgroupe(6, 1)
anigroupe(7).visible = True
anigroupe(8).Caption = tabgroupe(7, 1)
anigroupe(8).visible = True
anigroupe(9).Caption = tabgroupe(8, 1)
anigroupe(9).visible = True
tabtempgrp(1) = tabgroupe(1, 1)
tabtempgrp(2) = tabgroupe(2, 1)
tabtempgrp(3) = tabgroupe(3, 1)
tabtempgrp(4) = tabgroupe(4, 1)
tabtempgrp(6) = tabgroupe(5, 1)
tabtempgrp(7) = tabgroupe(6, 1)

```

```

    tabtempgrp(8) = tabgroupe(7, 1)
    tabtempgrp(9) = tabgroupe(8, 1)

    Case 9
    For i% = 1 To nbrgroupe
        anigroupe(i%).Caption = tabgroupe(i%, 1)
        anigroupe(i%).visible = True
        tabtempgrp(i%) = tabgroupe(i%, 1)
    Next i%
    End Select
    prochisto
Else
    afficher
    cursus.Show
End If
End If
Exit Sub
errorhandler:
Resume un
End Sub

Sub Form_Deactivate ()

picture1.visible = False
'Remet tous les boutons "groupes" invisibles
cmdprintcrs.visible = False
For i% = 1 To 9
    anigroupe(i%).visible = False
Next i%
For i% = 1 To 3
    lblintit(i%).visible = False
    lblintit(i%).Caption = ""
    lblprof(i%).visible = False
    lblprof(i%).Caption = ""
    lblheures(i%).visible = False
    lblheures(i%).Caption = ""
Next i%
line1.visible = False
scroll1.visible = False
scroll1.Enabled = False
line2.visible = False
line3.visible = False
line4.visible = False
line5.visible = False
command3d1.visible = False
command3d1.Enabled = False
command3d2.visible = False
command3d2.Enabled = False

End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Picture2_Click ()
guignols
End Sub

Sub scroll_Change ()
If boolaide = False Then
    lblintit(1) = tabintit(scroll1.Value + 1, 1)
    lblprof(1) = tabp(scroll1.Value + 1)
    lblheures(1) = tabh(scroll1.Value + 1)
    lblintit(2) = tabintit(scroll1.Value + 2, 1)
    lblprof(2) = tabp(scroll1.Value + 2)
    lblheures(2) = tabh(scroll1.Value + 2)
    lblintit(3) = tabintit(scroll1.Value + 3, 1)
    lblprof(3) = tabp(scroll1.Value + 3)
    lblheures(3) = tabh(scroll1.Value + 3)
ElseIf boolfin = True Then
    boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"

```


COURS.FRM - 8

End If
End Sub

Sub Timer1_Timer ()
cmdprintcrs.Enabled = True
End Sub

```

Sub CmdAnnée_Click (index As Integer)
If boolaide = False Then
    cu = index
    nbrimprloc = 0
    cours.Show
ElseIf boolfin = True Then
    boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

```

```

Sub Form_Activate ()
'Déclaration d'un dynaset pour le nombre d'années dans la faculté,
'et le nom de cette faculté
Dim ds_fac As dynaset

```

```

If nonfirst = True Then
    ecranecour = 12
    lbltitre.Caption = req_ville
    cursus.lblsstitre.Caption = tablbl(1, ecranecour)
    'Efface les boutons pour la première entrée dans la forme
    InitCursus
    'Accès BD pour affichage du logo universitaire
    MDI1!Data1.RecordSource = "select LOGO_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "' and
    NOM_V=''" + req_ville + "' and NOM_U=''" + req_univ + "'"
    MDI1!Data1.Refresh
    Set ds_unif = MDI1!Data1.Recordset
    If ds_unif("LOGO_U") <> "#NULL#" Then
        On Error GoTo errorhandler
        picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\" + "UNIV\" + ds_unif("LOGO_U") + ".bmp")
        picture1.Visible = True
    End If
un:
    'Accès BD pour affichage des boutons pour le choix de l'année
    MDI1!Data1.RecordSource = "select NOMBRE_A, NOM_F from FACULTE WHERE NOM_P = '" + req_pays + "'
    and NOM_V=''" + req_ville + "' and NOM_U=''" + req_univ + "'"
    MDI1!Data1.Refresh
    Set ds_fac = MDI1!Data1.Recordset
    If ds_fac.EOF = False Then
        how = ds_fac("NOMBRE_A")
        'Affichage des boutons
        For i% = 1 To ds_fac("NOMBRE_A")
            CmdAnnée(i%).Visible = True
            CmdAnnée(i%).Caption = tabannee(i%, 1)
        Next i%
        prochisto
    Else
        afficher
        menuiecs.Show
    End If
End If
Exit Sub
errorhandler:
Resume un
End Sub

```

```

Sub Form_Deactivate ()
picture1.Visible = False
For i% = 1 To 5
    cursus!CmdAnnée(i%).Visible = False
Next i%
End Sub

```

```

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

```

```

Sub Picture2_Click ()
guignols
End Sub

```


FBLANCHE.FRM - 1

```
Sub Timer1_Timer ()
NonFirst = True
boolfin = True
'Pour que VB reste toujours en avant plan.
Call setwindowpos(MDI1.hWnd, -1, 0, 0, 0, 0, &H10 Or &H40)
'Pour lancer la macro Print Screen
X = Shell(chem + "windows\RECORDER.EXE rec.rec", 1)
Pres.Show
Unload Me
End Sub
```

FHISTO.FRM - 1

Dim cp As Integer

Sub cmdhisto_Click (Index As Integer)

If boolaide = False Then

cp = 1

While cp <= cpthist - 1

cmdhisto(cp).Visible = False

cp = cp + 1

Wend

prechisto = False

req_pays = tabhisto(Index).country

req_ville = tabhisto(Index).city

req_univ = tabhisto(Index).univer

req_fac = tabhisto(Index).depart

l = tabhisto(Index).Langue

MDI1!Langue.Picture = LoadPicture(tabflag(1))

tabidform(tabhisto(Index).numero).Show

ElseIf boolfin = True Then

boolfin = False

MDI1.MM1.FileName = tabhlp2(1)

MDI1.MM1.Command = "open"

MDI1.MM1.Command = "play"

End If

End Sub

Sub Form_Activate ()

If nonfirst = True Then

initfhisto

tabprecform(24) = ecrancour

ecrancour = 24

prechisto = True

cp = 1

While cp <= cpthist - 1

cmdhisto(cp).Visible = True

cp = cp + 1

Wend

End If

End Sub

Sub Form_Unload (cancel As Integer)

cancel = True

End Sub


```

Sub cmdok_Click ()
If boolaide = False Then
    MenuVille.Show
ElseIf boolfin = True Then
    boolfin = False
MDI1.MMI.FileName = tabhlp2(1)
MDI1.MMI.Command = "open"
MDI1.MMI.Command = "play"
End If
End Sub

Sub Form_Activate ()
Dim ds_fac As dynaset
If nonfirst = True Then
    ecrancour = 21
    lbltitre.Caption = req_ville
    'Accès BD pour y prendre le nom de la ville, son logo en grand, la
    'photo présentée et la traduction de la phrase en 5 langues.
    MDI1!Data1.RecordSource = "select NOM_V, LOGO_V, PHOTO1_V, PHOTO2_V, PHRASEFRA_V, PHRASEANG_
    PHRASEALL_V, PHRASEESP_V, PHRASEITA_V from VILLE WHERE (NOM_P = '" + req_pays + "') and (NOM_V=
    + req_ville + "'"")
    MDI1!Data1.Refresh
    Set ds_Ville = MDI1!Data1.Recordset
    'Fin accès
    initintroville
    'Mise à jour des objets de la forme
    On Error GoTo errorhandler
    If ds_Ville("LOGO_V") <> "#NULL#" Then
        arret = "un"
        picture3.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(ds_
        le("NOM_V"), 8) + "\" + ds_Ville("LOGO_V") + ".bmp")
        picture3.Visible = True
    End If
un:
    If ds_Ville("PHOTO1_V") <> "#NULL#" Then
        arret = "deux"
        picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(ds_
        le("NOM_V"), 8) + "\" + ds_Ville("PHOTO1_V") + ".bmp")
        picture1.Visible = True
    End If
deux:
    If ds_Ville("PHOTO2_V") <> "#NULL#" Then
        arret = "trois"
        picture2.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(ds_
        le("NOM_V"), 8) + "\" + ds_Ville("PHOTO2_V") + ".bmp")
        picture2.Visible = True
    End If
trois:
    prochisto
Else
End If
Exit Sub

errorhandler:
Select Case arret
    Case "un"
        Resume un
    Case "deux"
        Resume deux
    Case "trois"
        Resume trois
End Select

End Sub

Sub Form_Deactivate ()
picture1.Visible = False
picture2.Visible = False
picture3.Visible = False
lblsstitre.Visible = False
End Sub

Sub Form_Unload (cancel As Integer)

```

INTROVIL.FRM - ,2

cancel = True
End Sub

Sub Picture4_Click ()
guignols
End Sub


```

Sub cmdunif_Click (index As Integer)
If boolaide = False Then
    req_univ = cmdunif(index).Caption
    introville.Show
ElseIf boolfin = True Then
    boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

```

```

Sub Form_Activate ()
If nonfirst = True Then
    'Accès BD pour y prendre le nom de la ville et le logo de la ville
    MDI1!Data1.RecordSource = "select NOM_V, LOGO_V from VILLE WHERE NOM_P = '" + req_pays + "'
d NOM_V='" + req_ville + "'"
    MDI1!Data1.Refresh
    If MDI1!Data1.Recordset("LOGO_V") <> "#NULL#" Then
        arret = "un"
        On Error GoTo errorhandler
        picture3.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\" + MDI1!Data1.Recordset("LOGO_V") + ".bmp")
        picture3.visible = True
    End If

```

```

un:
    ecrancour = 26
    'Pour gérer dynamiquement les retours en arrière au
    'départ de l'écran introville.
    tabprecform(21) = ecrancour

```

```

    lbltitre.Caption = req_ville
    MDI1!Data1.RecordSource = "Select NOM_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "' and N
_V='" + req_ville + "'"
    MDI1!Data1.Refresh
    Set ds_unif = MDI1!Data1.Recordset
    i% = 0
    Do While (Not ds_unif.EOF)
        cmdunif(i%).visible = True
        cmdunif(i%).Caption = ds_unif("NOM_U")
        ds_unif.MoveNext
        i% = i% + 1
    Loop

```

```

    Loop
    prochisto

```

```

End If
Exit Sub

```

```

errorhandler:
Select Case arret
Case "un"
    Resume un
End Select
End Sub

```

```

Sub Form_Deactivate ()
picture3.visible = False
End Sub

```

```

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

```

```

Sub Picture4_Click ()
guignols
End Sub

```

```

Sub CmdFrance_Click ()
pres.Timer1.Enabled = False
pres.Timer2.Enabled = False
pres.Timer3.Enabled = False
l = 1
MenuPrinc.Show
End Sub

```

```

Sub CmdGerm_Click ()
pres.Timer1.Enabled = False
pres.Timer2.Enabled = False
pres.Timer3.Enabled = False
l = 3
MenuPrinc.Show
End Sub

```

```

Sub CmdItaly_Click ()
pres.Timer1.Enabled = False
pres.Timer2.Enabled = False
pres.Timer3.Enabled = False
l = 4
MenuPrinc.Show
End Sub

```

```

Sub CmdSpain_Click ()
pres.Timer1.Enabled = False
pres.Timer2.Enabled = False
pres.Timer3.Enabled = False
l = 5
MenuPrinc.Show
End Sub

```

```

Sub CmdUK_Click ()
pres.Timer1.Enabled = False
pres.Timer2.Enabled = False
pres.Timer3.Enabled = False
l = 2
MenuPrinc.Show
End Sub

```

```

Sub printe_Click ()
If boolaide = False Then

ElseIf boolfin = True Then
boolfin = False
MM1.FileName = tabprnt(1)
MM1.Command = "open"
MM1.Command = "play"
End If
End Sub

```

```

Sub Aide_Click (Value As Integer)
If Value = True Then
    If ecrancour = 19 Then
        MenuUnifPart.cmdcont(0).Enabled = False
        MenuUnifPart.cmdcont(1).Enabled = False
        MenuUnifPart.cmdcont(2).Enabled = False
    End If
    boolaide = True
    If boolfin = False Then
        MM1.Command = "stop"
        MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MM1.Notify = False
    MM1.Wait = True
    MM1.FileName = tabhlp1(1)
    MM1.Command = "open"
    MM1.Command = "play"
Else
    If ecrancour = 19 Then
        MenuUnifPart.cmdcont(0).Enabled = True

```

```

MenuUnifPart.cmdcont(1).Enabled = True
MenuUnifPart.cmdcont(2).Enabled = True
End If
boolaide = False
If boolfin = False Then
MM1.Command = "stop"
MM1.Command = "close"
End If
End If
End Sub

Sub Histo_Click ()
If boolaide = False Then
'MsgBox "Ce bouton permettra de retourner à un des 9 derniers écrans qui ont été consultés.", 64,
Information"
fhisto.Show
Else
If boolfin = False Then
MM1.Command = "stop"
MM1.Command = "close"
End If
DoEvents
boolfin = False
MM1.FileName = tabhist(1)
MM1.Command = "open"
MM1.Command = "play"
End If
End Sub

Sub Langue_Click ()

If boolaide = False Then
'Changement du booléen
chgt1 = True
'Changement de la variable
l = l + 1
If l = nbl + 1 Then
l = 1
End If

'changement du drapeau
Langue.Picture = LoadPicture(tabflag(1))

'Traduction
Select Case MDI1.ActiveForm.Caption
Case "fhisto"
initfhisto
Case "MenuPays"
initmenupays
Case "MenuPrinc"
initmenuprinc
Case "MenuVille"
initmenuville
Case "MenuUnif"
initmenuunif
Case "Presunif1"
initpresunif1
Case "Stat1"
initstat1cl
Case "MenuIECS"
initmenuiecs
Case "cursus"
InitCursus
Case "Cours"
initcours
Case "orga"
InitOrgabis
Case "CondAdm"
InitCondAdm
Case "MenuInfos"
InitMenuInfos
Case "MnuAUt1"
InitMnuAut1
Case "MnuAUt2"
InitMnuAut2

```



```

    Case "MenuVieQuot"
    InitMenuVieQuot
    Case "MenuUnifPart"
    initMenuUnifPart
    Case "IntroVille"
    initintroville
    Case "Plans"
    InitPlans
    Case "Calend"
    InitCalend
    Case "AUBiblio"
    'Pas initbiblio, pour ne pas refaire les accès à la BD
    initbibliobis
    Case "ChgtUniv"
    initchgtunivbis
    Case Else
End Select

Else
    If boolfin = False Then
        MM1.Command = "stop"
        MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MM1.FileName = tablang(1)
    MM1.Command = "open"
    MM1.Command = "play"
End If

End Sub

Sub MDIForm_Deactivate ()
Me.Show
End Sub

Sub MDIForm_Load ()
initialisation

ListForm
chargement
'Remplissage du dynaset ds contenant tous les pays présents
'dans la Base de données
data1.RecordSource = "select NOM_P from PAYS"
data1.Refresh
Set ds = data1.Recordset
data1.Recordset.MoveLast
nbrpays = data1.Recordset.RecordCount
data1.Recordset.MoveFirst
'Do While Not data1.Recordset.EOF ' ds.EOF
'    labell.Caption = data1.Recordset("NOM_P") ' ds("nom")
'    data1.Recordset.MoveNext ' ds.MoveNext
'Loop

FBlanche.Show
End Sub

Sub MDIForm_Resize ()
Me.WindowState = 2
End Sub

Sub MDIForm_Unload (cancel As Integer)
cancel = True
End Sub

Sub Menu_Click ()
Dim cp As Integer

If boolaide = False Then
    If ecrancour = 24 Then
        cp = 1
        While cp <= cpthist - 1

```

```

        fhisto.cmdhisto(cp).Visible = False
        cp = cp + 1
    Wend
End If
MenuPrinc.Show
Else
    If boolfin = False Then
        MM1.Command = "stop"
        MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MM1.FileName = tabmenu(1)
    MM1.Command = "open"
    MM1.Command = "play"
End If
End Sub

Sub MM1_Done (NotifyCode As Integer)
boolfin = True
MM1.Command = "close"
End Sub

Sub quitter_Click ()
Dim cp As Integer

If boolaide = False Then
    pres.Show
Else
    If boolfin = False Then
        MM1.Command = "stop"
        MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MM1.FileName = tabquit(1)
    MM1.Command = "open"
    MM1.Command = "play"
End If
End Sub

Sub Retour_Click ()
Dim cp As Integer

If boolaide = False Then
    If (ecrancour = 13) And (groupe = False) Then
        cours!Cmdprintcrs.Visible = True
        cours!scroll.Value = 0
        For i% = 1 To 3
            cours!lblprof(i%).Visible = False
            cours!lblintit(i%).Visible = False
            cours!lblheures(i%).Visible = False
            cours!scroll.Visible = False
            cours!scroll.Enabled = False
            cours!Command3D1.Visible = False
            cours!Command3D1.Enabled = False
            cours!Command3D2.Visible = False
            cours!Command3D2.Enabled = False
        Next i%
        cours!Line1.Visible = False
        cours!Line2.Visible = False
        cours!Line3.Visible = False
        cours!Line4.Visible = False
        cours!Line5.Visible = False
        groupe = True
        initcours
    Else
        If ecranccour = 24 Then
            cp = 1
            While cp <= cpthist - 1
                fhisto.cmdhisto(cp).Visible = False
                cp = cp + 1
            Wend
        End If
    End If

```

```
        tabidform(tabprecform(ecrancour)).Show
    End If
Else
    If boolfin = False Then
        MM1.Command = "stop"
        MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MM1.FileName = tabreto(1)
    MM1.Command = "open"
    MM1.Command = "play"
End If
End Sub

Sub Unif_Click ()
    If boolaide = False Then
        ChgtUniv.Show
        MsgBox "Ce bouton permettra de changer d'université.", 64, "Information"
    Else
        If boolfin = False Then
            MM1.Command = "stop"
            MM1.Command = "close"
        End If
        DoEvents
        boolfin = False
        MM1.FileName = tabunif(1)
        MM1.Command = "open"
        MM1.Command = "play"
    End If
End Sub
```



```

Sub cmdcondad_Click ()
If boolaide = False Then
condadm.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = chem + "telemaqu\son\help2.wav"
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub

```

```

Sub cmdcurs_Click ()
If boolaide = False Then
cursus.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = tabhlp2(1)
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub

```

```

Sub cmdorga_Click ()
If boolaide = False Then
orga.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = chem + "telemaqu\son\help2.wav"
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub

```

```

Sub Form_Activate ()
If nonfirst = True Then
'Pour remettre la barre d'icônes visible après la présentation
mdil.Picture1.Visible = True

```

```

mdil!Data1.RecordSource = "select NOM_F from FACULTE WHERE NOM_P = '" + req_pays + "' and NO
V=''" + req_ville + "'" and NOM_U=''" + req_univ + "'"
mdil!Data1.Refresh
lbltitre.Caption = req_ville
If mdil!Data1.Recordset.EOF = False Then
cmdcurs.Visible = True
cmdorga.Visible = True
cmdcondad.Visible = True
lblsstitre.Caption = mdil!Data1.Recordset("NOM_F")
cmdcondad.Visible = (req_univ = "IECS Strasbourg")
ecrancour = 5
initmenuiecs
mdil!Data1.RecordSource = "select LOGO_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "'
and NOM_V=''" + req_ville + "'" and NOM_U=''" + req_univ + "'"
mdil!Data1.Refresh
Set ds_unif = mdil!Data1.Recordset
If ds_unif("LOGO_U") <> "#NULL#" Then
On Error GoTo errorhandler
Picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(
q_ville, 8) + "\" + "UNIV\" + ds_unif("LOGO_U") + ".bmp")
Picture1.Visible = True
End If
un:
prochisto
Else
lblsstitre.Caption = tabdepiecs(1)
afficher
MenuUnif.Show
End If
End If
Exit Sub
errorhandler:

```

MENUIECS.FRM - 2

Resume un:
End Sub

Sub Form_Deactivate ()
Picture1.Visible = False
cmdcurs.Visible = False
cmdorga.Visible = False
cmdcondad.Visible = False
End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Picture2_Click ()
guignols
End Sub

```

Sub CmdAdrUt_Click ()
If boolaide = False Then
MnuAUT1.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

Sub Cmdcalend_Click ()
Dim i As Integer

If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabCalend(i)
Next
Aubiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

Sub CmdPlan_Click ()
Dim i As Integer

If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabPlan(i)
Next
tabprecform(15) = ecrancour
Plans.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

Sub CmdVieQuot_Click ()
If boolaide = False Then
MenuVieQuot.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub

Sub Form_Activate ()
If nonfirst = True Then
    ecrancour = 6
    tabprecform(23) = ecrancour
    lbltitre.Caption = req_ville
    InitMenuInfos
    prochisto
Else
End If
End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

```


MENUINFO.FRM - 2.

```
Sub Picture1_Click ()  
guignols  
End Sub
```

MENUPART.FRM - 1

Dim avant As Integer

Sub carte_MouseDown (index As Integer, Button As Integer, Shift As Integer, x As Single, y As Single)

'label1.Caption = Str(x) + Str(y)

'Variable pour condition de sortie dans la boucle de
'comparaison entre ds des pays de la bd et du nom de pays
'sur lequel on vient de cliquer
trouve% = False

'Remplissage du dynaset ds contenant tous les pays présents
'dans la Base de données (redondant mais utile si on vient
'de l'écran suivant)

mdil!Data1.RecordSource = "select NOM_P from PAYS"

mdil!Data1.Refresh

Set ds = mdil!Data1.Recordset

'All=1, Pays_Bas=2, Belg=3, Dane=4,finlande=5, sued=6

'G-B =7, Irlande=8, canada=9,USA=10, Grèce=11,Autriche=12

'Suisse=13, Italie=14, Espagne=15

If boolaide = False Then

cmdok.Visible = False

Select Case index

Case 0

If (x > 780) And (x < 1485) And (y > 2430) And (y < 3225) Then

'Le pays appartient-il à la BD?

Do While (Not ds.EOF) And (trouve% = faux)

If ds("NOM_P") = tabenspays(8, 1) Then

trouve% = True

lblpays.Caption = tabenspays(8, 1)

pays_ok = 8

End If

ds.MoveNext

Loop

If trouve% = False Then

'Pour forcer la référence à Tabenspays sur sa valeur blanche.

pays_ok = 0

lblpays.Caption = ""

afficher

End If

ElseIf (x > 1486) And (x < 1695) And (y > 2475) And (y < 2790) Then

'Le pays appartient-il à la BD?

Do While (Not ds.EOF) And (trouve% = faux)

If ds("NOM_P") = tabenspays(7, 1) Then

trouve% = True

lblpays.Caption = tabenspays(7, 1)

pays_ok = 7

End If

ds.MoveNext

Loop

If trouve% = False Then

pays_ok = 0

lblpays.Caption = ""

afficher

End If

ElseIf (x > 1665) And (x < 2490) And (y > 1785) And (y < 3855) Then

'Le pays appartient-il à la BD?

Do While (Not ds.EOF) And (trouve% = faux)

If ds("NOM_P") = tabenspays(7, 1) Then

trouve% = True

lblpays.Caption = tabenspays(7, 1)

pays_ok = 7

End If

ds.MoveNext

Loop

If trouve% = False Then

pays_ok = 0

lblpays.Caption = ""

afficher

End If

```

ElseIf (x > 4320) And (x < 5340) And (y > 0) And (y < 3105) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(6, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(6, 1)
            pays_ok = 6
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 5415) And (x < 6420) And (y > 0) And (y < 2070) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(5, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(5, 1)
            pays_ok = 5
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 3555) And (x < 4035) And (y > 2505) And (y < 3165) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(4, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(4, 1)
            pays_ok = 4
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 2520) And (x < 3135) And (y > 3855) And (y < 4410) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(3, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(3, 1)
            pays_ok = 3
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 2745) And (x < 3315) And (y > 3360) And (y < 3856) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(2, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(2, 1)
            pays_ok = 2
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

```



```

        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 3316) And (x < 4455) And (y > 3166) And (y < 4425) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(1, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(1, 1)
            pays_ok = 1
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 990) And (x < 2000) And (y > 30) And (y < 960) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(16, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(16, 1)
            pays_ok = 16
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        'Pour forcer la référence à Tabenspays sur sa valeur blanche.
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 3330) And (x < 4320) And (y > 0) And (y < 2400) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(17, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(17, 1)
            pays_ok = 17
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        'Pour forcer la référence à Tabenspays sur sa valeur blanche.
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 4455) And (x < 5895) And (y > 3240) And (y < 4425) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(18, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(18, 1)
            pays_ok = 18
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        'Pour forcer la référence à Tabenspays sur sa valeur blanche.
        pays_ok = 0

```

```

        lblpays.Caption = ""
        afficher
    End If

Else
    lblpays.Caption = ""
    cmdok.Visible = False
End If

Case 1
If (x > 525) And (x < 2085) And (y > 1545) And (y < 3255) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(15, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(15, 1)
            pays_ok = 15

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 3270) And (x < 4350) And (y > 1350) And (y < 2385) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(14, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(14, 1)
            pays_ok = 14

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 3975) And (x < 5070) And (y > 2386) And (y < 3420) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(14, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(14, 1)
            pays_ok = 14

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 3135) And (x < 3945) And (y > 975) And (y < 1515) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(13, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(13, 1)
            pays_ok = 13

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""

```

```

        afficher
    End If

ElseIf (x > 3900) And (x < 4245) And (y > 1110) And (y < 1350) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(12, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(12, 1)
            pays_ok = 12
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 4244) And (x < 4980) And (y > 1005) And (y < 1470) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(12, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(12, 1)
            pays_ok = 12
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 5775) And (x < 6810) And (y > 2760) And (y < 4320) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(11, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(11, 1)
            pays_ok = 11
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 0) And (x < 525) And (y > 1635) And (y < 2880) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(19, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(19, 1)
            pays_ok = 19
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 4305) And (x < 6075) And (y > 360) And (y < 1005) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(20, 1) Then
            trouve% = True

```



```

        lblpays.Caption = tabenspays(20, 1)
        pays_ok = 20

        End If
    ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 4980) And (x < 6015) And (y > 1050) And (y < 1605) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(21, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(21, 1)
            pays_ok = 21

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 6015) And (x < 7185) And (y > 1065) And (y < 2085) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(22, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(22, 1)
            pays_ok = 22

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 6060) And (x < 7080) And (y > 2085) And (y < 2760) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(23, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(23, 1)
            pays_ok = 23

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 4470) And (x < 6060) And (y > 1605) And (y < 2595) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM_P") = tabenspays(24, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(24, 1)
            pays_ok = 24

            End If
        ds.MoveNext
    Loop
    If trouve% = False Then

```

```

        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

ElseIf (x > 5400) And (x < 5775) And (y > 2595) And (y < 3200) Then
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(25, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(25, 1)
            pays_ok = 25
        End If
        ds.MoveNext
    Loop
    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If

Else
    lblpays.Caption = ""
    cmdok.Visible = False
End If

Case 2
If (x > 2760) And (x < 3840) And (y > 495) And (y < 1920) Then
    lblpays.Caption = ""
    cmdok.Visible = False

Else
    'Le pays appartient-il à la BD?
    Do While (Not ds.EOF) And (trouve% = faux)
        If ds("NOM P") = tabenspays(9, 1) Then
            trouve% = True
            lblpays.Caption = tabenspays(9, 1)
            pays_ok = 9
        End If
        ds.MoveNext
    Loop

    If trouve% = False Then
        pays_ok = 0
        lblpays.Caption = ""
        afficher
    End If
End If

End Select

If trouve% = True Then
    cmdok.Visible = True
End If
req_pays = tabenspays(pays_ok, 1)

ElseIf boolfin = True Then
    boolfin = False
    mdil.MM1.FileName = tabhlp2(1)
    mdil.MM1.Command = "open"
    mdil.MM1.Command = "play"
End If
End Sub

Sub cmdcont_Click (index As Integer, Value As Integer)
    If Value = True Then
        lblchoixpays.Visible = True
        lblpc.Visible = True
        lblpays.Visible = True
        carte(avant).Visible = False
        carte(index).Visible = True
        avant = index
    End If
End Sub

```

```

Else
    'Pour ne pas laisser la référence vers un élément de Tabenspays
    pays_ok = 0
    carte(index).Visible = False
    lblchoixpays.Visible = False
    lblpc.Visible = False
End If
lblpays.Caption = ""
cmdok.Visible = False
End Sub

Sub cmdok Click ()
If boolaide = False Then
    pays = lblpays.Caption
    MenuPays.Show
ElseIf boolfin = True Then
    boolfin = False
    mdi1.MM1.FileName = tabhlp2(1)
    mdi1.MM1.Command = "open"
    mdi1.MM1.Command = "play"
End If
End Sub

Sub Form_Activate ()
Dim p As Integer
'Variable pour les changements de langue
chgtl = False

If nonfirst = True Then
    EcranCour = 19
    initMenuUnifPart
    For p% = 0 To 2
        cmdcont(p).Value = False
    Next p%
    prochisto
End If
End Sub

Sub Form_Deactivate ()

For p = 0 To 2
    carte(p).Visible = False
Next
lblchoixpays.Visible = False
lblpc.Visible = False
lblpays.Visible = False
lblpays.Caption = ""
cmdok.Visible = False

End Sub

Sub Form_MouseDown (Button As Integer, Shift As Integer, x As Single, y As Single)
If (boolaide = True) And (boolfin = True) Then
    If (x > 8040) And (x < 11415) And (y > 2040) And (y < 6495) Then
        boolfin = False
        mdi1.MM1.FileName = tabhlp2(1)
        mdi1.MM1.Command = "open"
        mdi1.MM1.Command = "play"
    End If
End If
End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Picture1_Click ()
guignols
End Sub

```


MENUPAYS.FRM - 1

```
Const nbre = 8
Dim tabnomville() As String
Dim cpville, cp, compteur As Integer
Dim numdrap As Integer

Sub CmdVille_Click (index As Integer)
If boolaide = False Then
'Enlève le numero et le point devant le nom de la ville
Dim k As Integer
k = 1
While (Mid$(cmdville(index).Caption, k, 1) >= "0" And Mid$(cmdville(index).Caption, k, 1) <= "9")
r (Mid$(cmdville(index).Caption, k, 1) = " ") Or (Mid$(cmdville(index).Caption, k, 1) = ".")
k = k + 1
Wend
req_ville = Mid$(cmdville(index).Caption, k, (Len(cmdville(index).Caption) - (k - 1)))

'Accès BD pour savoir combien d'université(s) il y a pour la ville req_ville
MDI1!Data1.RecordSource = "Select NOM_U from UNIVERSITE WHERE NOM_P ='" + req_pays + "' and NOM_V
" + req_ville + "'"
MDI1!Data1.Refresh
Set ds_unif = MDI1!Data1.Recordset

If ds_unif.EOF <> True Then
'On compte combien d'unif il y a dans la ville
ds_unif.MoveLast
combien& = ds_unif.RecordCount
ds_unif.MoveFirst
'Test pour savoir s'il y a plus d'une université par ville
'Si oui, on va sur IntroVilleBis
If combien& = 1 Then
introville.Show
req_univ = ds_unif("NOM_U")
Else
IntrovilleBis.Show
End If
Else
introville.Show
req_univ = ""
End If

ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If

End Sub

Sub Form_Activate ()
If nonfirst = True Then
ecrancour = 20
'Pour gérer dynamiquement les retours en arrière au
'départ de l'écran introville.
tabprecform(21) = ecrancour

lbltitre.Caption = pays
'reinitialisation des boutons
For i% = 1 To 8
cmdville(i%).Caption = ""
Next i%
'Accès à la BD pour y prendre la carte du
'pays et son drapeau
MDI1!Data1.RecordSource = "select CARTE_P, DRAPEAU_P from pays where NOM_P ='" + req_pays + "'
'pays
MDI1!Data1.Refresh
Set ds = MDI1!Data1.Recordset
If ds("CARTE_P") <> "#NULL#" Then
On Error GoTo ErrorHandler
arret = "un"
picture2.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + ds("CARTE_
) + ".bmp")
picture2.Visible = True
```

```

End If
un:
  If ds("DRAPEAU P") <> "#NULL#" Then
    On Error GoTo ErrorHandler
    arret = "deux"
    If (pays_ok = 3) Or (pays_ok = 20) Or (pays_ok = 24) Then
      picclip1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + ds("DR
EAU_P") + ".bmp")
      picture1.Picture = picclip1.GraphicCell(0)
      picture1.Visible = True
      numdrap = 0
      timer1.enabled = True
    Else
      picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + ds("DR
EAU_P") + ".bmp")
      picture1.Visible = True
    End If
  End If
deux:
  'Accès à la BD pour y prendre les villes et leurs numeros
  MDI1!Data1.RecordSource = "select NUMERO_V, NOM_V from VILLE WHERE NOM_P = '" + req_pays + "'
RDER BY NUMERO V"
  MDI1!Data1.Refresh
  Set ds_ville = MDI1!Data1.Recordset
  If ds_ville.EOF = False Then
    ds_ville.MoveLast
    ReDim tabnomville(ds_ville.RecordCount)
    ds_ville.MoveFirst
    cpville = 1
    Do While Not ds_ville.EOF
      tabnomville(cpville) = Str(ds_ville("NUMERO_V")) + ". " + ds_ville("NOM_V")
      ds_ville.MoveNext
      cpville = cpville + 1
    Loop

    cp = 1
    compteur = 1
    While (cp <= UBound(tabnomville)) And (compteur <= nbre)
      cmdville(compteur).Caption = tabnomville(cp)
      cmdville(compteur).Visible = True
      cp = cp + 1
      compteur = compteur + 1
    Wend
    If cp <= UBound(tabnomville) Then
      suiv.Visible = True
    End If
    cp = cp - compteur + 1

    prochisto
  Else
    afficher
    MenuUnifPart.Show
  End If
End If

Exit Sub ' Exit before entering
ErrorHandler:
Select Case arret
Case "un"
  Resume un
Case "deux"
  Resume deux
End Select

End Sub

Sub Form_Deactivate ()
picture1.Visible = False
picture2.Visible = False
For i% = 1 To 8
  cmdville(i%).Caption = ""
  cmdville(i%).Visible = False
Next i%

```

MENUPAYS.FRM - 3

```
timer1.enabled = False
End Sub
```

```
Sub Form_Unload (cancel As Integer)
cancel = True
End Sub
```

```
Sub Picture4_Click ()
guignols
End Sub
```

```
Sub prec_Click ()
If boolaide = False Then
    suiv.Visible = True
    compteur = 1
    While compteur <= nbre
        cmdville(compteur).Visible = False
        compteur = compteur + 1
    Wend
    compteur = 1
    cp = cp - nbre
    If cp <= 1 Then
        prec.Visible = False
    End If
    While compteur <= nbre
        cmdville(compteur).Caption = tabnomville(cp)
        cmdville(compteur).Visible = True
        cp = cp + 1
        compteur = compteur + 1
    Wend
    cp = cp - nbre
End Sub
```

```
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabhlp2(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub suiv_Click ()
If boolaide = False Then
    prec.Visible = True
    compteur = 1
    While compteur <= nbre
        cmdville(compteur).Visible = False
        compteur = compteur + 1
    Wend
    compteur = 1
    cp = cp + nbre
    While (cp <= UBound(tabnomville)) And (compteur <= nbre)
        cmdville(compteur).Caption = tabnomville(cp)
        cmdville(compteur).Visible = True
        cp = cp + 1
        compteur = compteur + 1
    Wend
    If cp > UBound(tabnomville) Then
        suiv.Visible = False
    End If
    cp = cp - compteur + 1
End Sub
```

```
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabhlp2(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
```


MENUPAYS.FRM - 4

End Sub

Sub Timer1_Timer ()

If numdrap = 0 Then

 picture1.Picture = picclip1.GraphicCell(1)

 numdrap = 1

Else

 picture1.Picture = picclip1.GraphicCell(0)

 numdrap = 0

End If

End Sub

MENUPRIN.FRM - 1

```
Sub cmdIECS_Click ()
If boolaide = False Then
req_pays = "France"
req_ville = "Strasbourg"
req_univ = "IECS Strasbourg"
req_fac = "IECS"
menuville.Show
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
```

End Sub

```
Sub cmdUnifPart_Click ()
If boolaide = False Then
MenuUnifPart.Show
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
```

End Sub

```
Sub Form_Activate ()
If nonfirst = True Then
ecrancour = 2
initmenuprinc
MDI1.Langue.Picture = LoadPicture(tabflag(1))
MDI1.Picture1.Visible = True
prochisto
```

```
Else
End If
End Sub
```

```
Sub Picture3_Click ()
guignols
End Sub
```

```
Sub cmdchiffres_Click ()
If boolaide = False Then
stat1.Show
'Unload Me
```

```
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = tabhlp2(1)
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub
```

```
Sub cmddepieces_Click ()
If boolaide = False Then
menuiecs.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = tabhlp2(1)
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub
```

```
Sub cmdpresunif_Click ()
If boolaide = False Then
presunif1.Show
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = tabhlp2(1)
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub
```

```
Sub Form_Activate ()
If nonfirst = True Then
ecrancour = 4
initmenuunif
lbltitre.Caption = req_ville
If req_univ <> "" Then
cmdpresunif.Visible = True
cmdchiffres.Visible = True
cmddepieces.Visible = True
'Pour remettre la barre d'icônes visible après la présentation
mdil.Picture1.Visible = True
mdil!Data1.RecordSource = "select LOGO_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "
and NOM_V=''" + req_ville + "'" and NOM_U=''" + req_univ + "'"
mdil!Data1.Refresh
Set ds_unif = mdil!Data1.Recordset
If ds_unif("LOGO_U") <> "#NULL#" Then
On Error GoTo errorhandler
Picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(
q_ville, 8) + "\UNIV\" + ds_unif("LOGO_U") + ".bmp")
Picture1.Visible = True
End If
un:
'Accès BD pour prendre le nom de la faculté
mdil!Data1.RecordSource = "select NOM_F from FACULTE where (NOM_P = '" + req_pays + "'" and
(NOM_V= '" + req_ville + "'" and (NOM_U=''" + req_univ + "'"
mdil!Data1.Refresh
If mdil!Data1.Recordset.EOF = False Then
req_fac = mdil!Data1.Recordset("NOM_F")
End If
prochisto
Else
afficher
Menuville.Show
End If
End If
Exit Sub
errorhandler:
```


MENUUNIF.FRM - 2

Resume un
End Sub

Sub Form_Deactivate ()
Picture1.Visible = False
cmdpresunif.Visible = False
cmdchiffres.Visible = False
cmddepieces.Visible = False
End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Picture2_Click ()
guignols
End Sub

MENUVIEQ.FRM - 1

Dim i As Integer

```
Sub CmdVQBanques_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabVQBanques(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdVQCinTh_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabVQCinTh(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdVQLog_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabVQLog(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdVQPostes_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabVQPostes(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdVQRestaus_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabVQRestaus(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
```

MENUVIEQ.FRM - 2

End Sub

```
Sub CmdVQSport_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabVQSport(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub Form_Activate ()
If nonfirst = True Then
    ecrancour = 7
    tabprecform(23) = ecrancour
    lbltitre.Caption = req_ville
    InitMenuVieQuot
    prochisto
Else
End If
End Sub
```

```
Sub Form_Unload (cancel As Integer)
cancel = True
End Sub
```

```
Sub Picture1_Click ()
guignols
End Sub
```



```
Sub cmdInfPrat_Click ()
If boolaide = False Then
MenuInfos.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = tabhlp2(1)
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub
```

```
Sub cmdInfUnif_Click ()
If boolaide = False Then
MenuUnif.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = tabhlp2(1)
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub
```

```
Sub cmdPresVille_Click ()

If boolaide = False Then
Presville.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
mdil.MM1.FileName = tabhlp2(1)
mdil.MM1.Command = "open"
mdil.MM1.Command = "play"
End If
End Sub
```

```
Sub Form_Activate ()
If nonfirst = True Then
If req_univ = "IECS Strasbourg" Then
tabprecform(3) = 2
Else
tabprecform(3) = 21
End If
ecrancour = 3
initmenuville
lbtitre.Caption = req_ville
'Pour remettre la barre d'icônes visible après la présentation
mdil.picture1.Visible = True
'Accès BD pour y prendre le nom de la ville et le logo de la ville
mdil!Data1.RecordSource = "select NOM_V, LOGO_V from VILLE WHERE NOM_P = '" + req_pays + "'
d NOM V='" + req_ville + "'"
mdil!Data1.Refresh
Set ds_Ville = mdil!Data1.Recordset
If ds_Ville("LOGO_V") <> "#NULL#" Then
On Error GoTo errorhandler
arret = "un"
picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\" + ds_Ville("LOGO_V") + ".bmp")
picture1.Visible = True
End If
un:
prochisto
End If
Exit Sub
errorhandler:
Select Case arret
Case "un"
Resume un
End Select

End Sub
```

MENUVILL.FRM - 2

```
Sub Form_Deactivate ()  
picture1.Visible = False  
End Sub
```

```
Sub Form_Unload (cancel As Integer)  
cancel = True  
End Sub
```

```
Sub Picture2_Click ()  
guignols  
End Sub
```

Dim i As Integer

```
Sub CmdAUAssocEt_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabAUAssocEt(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdAUBiblio_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabAUBiblio(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdAUCentre_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabAUCentre(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub cmdAUSecr_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabAUSecr(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub CmdAUUnif_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabAUUnif(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
```


MNUAUT1.FRM - 2

End Sub

```
Sub CmdCinTh2_Click ()
If boolaide = False Then
For i = 1 To nbl
sstitre(i) = tabAUCentre2(i)
Next
AUBiblio.Show
'Unload Me
ElseIf boolfin = True Then
boolfin = False
MDI1.MM1.FileName = tabhlp2(1)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End If
End Sub
```

```
Sub Form_Activate ()
If nonfirst = True Then
    ecrancour = 8
    tabprecform(23) = ecrancour
    lbltitre.Caption = req_ville
    InitMnuAut1
    prochisto
Else
```

```
End If
End Sub
```

```
Sub Form_Unload (cancel As Integer)
cancel = True
End Sub
```

```
Sub Picture1_Click ()
guignols
End Sub
```

ORGA.FRM - 1

Const nbre = 4

```
Sub CmdPrint_Click ()
Dim strtitre As String
Dim nbrp As Integer
```

```
If boolaide = False Then
cmdprint.Enabled = False
timer1.Enabled = True
If improk() = True Then
    On Error GoTo errhandler
    printer.ScaleMode = 7
    compteur = 0
    If (UBound(taborgan) Mod nbritems = 0) Then
        nbrp = UBound(taborgan) / nbritems
    Else
        nbrp = (UBound(taborgan) \ nbritems) + 1
    End If
    printer.FontName = "Times New Roman"
    printer.FontSize = 30
    printer.FontBold = True
    strtitre = req_fac + " : " + taborga(1)
    printer.CurrentX = (printer.Width / 567 - Len(strtitre) * .5) / 2
    printer.Print (strtitre)
    printer.ScaleLeft = -2
    printer.Print
    printer.FontSize = 14
    printer.FontBold = False
    While compteur < UBound(taborgan)
        compteur = compteur + 1
        printer.FontItalic = True
        printer.Print taborgafct(compteur, 1)
        printer.FontItalic = False
        printer.Print taborgan(compteur)
        printer.Print
        If (compteur Mod nbritems = 0) And (compteur <> UBound(taborgan)) Then
            printer.CurrentX = 16
            printer.CurrentY = 26
            printer.Print (printer.Page + "/" + Str(nbrp))
            printer.NewPage
            printer.ScaleLeft = 0
            printer.FontSize = 30
            printer.FontBold = True
            printer.CurrentX = (printer.Width / 567 - Len(strtitre) * .5) / 2
            printer.Print (strtitre)
            printer.ScaleLeft = -2
            printer.Print
            printer.FontSize = 14
        End If
    Wend
    printer.CurrentX = 16
    printer.CurrentY = 26
    printer.Print (printer.Page + "/" + Str(nbrp))
    printer.EndDoc
End If
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabprnt(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
fin: Exit Sub
errhandler:
Resume fin

End Sub

Sub Form_Activate ()
If nonfirst = True Then
    EcranCour = 14
```

ORGA.FRM - 2

```
    lbltitre.Caption = req_ville
    lblsstitre.Caption = taborga(1)
    InitOrga
    nbrimprloc = 0
    prochisto
End If
End Sub
```

Sub Form_Deactivate ()

```
For h% = 0 To 3
    lblfonc(h%).visible = False
    lblnom(h%).visible = False
Next h%
prec.visible = False
suiv.visible = False
picture1.visible = False
End Sub
```

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Picture2_Click ()
guignols
End Sub

```
Sub prec_Click ()
If boolaide = False Then
    suiv.visible = True
    compteur = 0
    While compteur < nbre
        lblnom(compteur).visible = False
        lblfonc(compteur).visible = False
        compteur = compteur + 1
    Wend
    compteur = 0
    cporga = cporga - nbre
    chl = cporga
    If cporga <= 1 Then 'LBound(tabinitbiblio) Then
        prec.visible = False
    End If
    While compteur < nbre
        lblnom(compteur).Caption = taborgan(cporga)
        lblfonc(compteur).Caption = taborgafct(cporga, 1)
        lblfonc(compteur).visible = True
        lblnom(compteur).visible = True
        cporga = cporga + 1
        compteur = compteur + 1
    Wend
    cporga = cporga - nbre
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabhlp2(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
```

End Sub

```
Sub suiv_Click ()
If boolaide = False Then
    prec.visible = True
    compteur = 0
    While compteur < nbre
        lblnom(compteur).visible = False
        lblfonc(compteur).visible = False
        compteur = compteur + 1
    End While
End Sub
```



```

Wend
compteur = 0
cporga = cporga + nbre
chl = cporga
While (cporga <= nbrorga) And (compteur < nbre)
    lblnom(compteur).Caption = taborgan(cporga)
    lblfonc(compteur).Caption = taborgafct(cporga, 1)
    lblfonc(compteur).visible = True
    lblnom(compteur).visible = True
    cporga = cporga + 1
    compteur = compteur + 1
Wend
If cporga > nbrorga Then
    suiv.visible = False
End If
cporga = cporga - (compteur)
Else
    If boolfin = False Then
        MDI1.MM1.Command = "stop"
        MDI1.MM1.Command = "close"
    End If
    DoEvents
    boolfin = False
    MDI1.MM1.FileName = tabhlp2(1)
    MDI1.MM1.Command = "open"
    MDI1.MM1.Command = "play"
End If
End Sub

Sub Timer1_Timer ()
cmdprint.Enabled = True
End Sub

```

```

Sub Form_Activate ()
If nonfirst = True Then
    Dim ds_plan As dynaset
    ecrancour = 15
    lbltitre = req_ville
    InitPlans
    'Accès BD pour y prendre le logo et le plan
    mdil!Data1.RecordSource = "select PLAN_V from VILLE WHERE (NOM_P = '" + req_pays + "') and (I
M_V= '" + req_ville + "')"
    mdil!Data1.Refresh
    Set ds_plan = mdil!Data1.Recordset
    'Fin accès

    If ds_plan("PLAN_V") <> "#NULL#" Then
        On Error GoTo errorhandler1
        picture3.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\" + ds_plan("PLAN_V") + ".bmp")
        picture3.Visible = True
        prochisto
    Else
        afficher
        tabidform(tabprecform(ecrancour)).Show
    End If

un:
End If
Exit Sub
errorhandler1:
Resume un
End Sub

Sub Form_Deactivate ()
picture2.Visible = False
picture3.Visible = False
End Sub

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

Sub Picture1_Click ()
guignols
End Sub

```

PRES.FRM - 1

```
Const c1 = 50
Const c2 = 50
Const c3 = 50
Dim l1, t1, l2, t2, l3, t3, timecompt As Integer
```

```
Sub Command3D1_Click ()
l = 5
MDI1.Show
End Sub
```

```
Sub Command3D2_Click ()
l = 4
MDI1.Show
End Sub
```

```
Sub Command3D3_Click ()
l = 1
MDI1.Show
End Sub
```

```
Sub Command3D4_Click ()
l = 2
MDI1.Show
End Sub
```

```
Sub Command3D5_Click ()
l = 3
MDI1.Show
End Sub
```

```
Sub CmdFrance_Click ()
```

```
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
l = 1
```

```
MenuPrinc.Show
End Sub
```

```
Sub CmdGerm_Click ()
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
l = 3
MenuPrinc.Show
End Sub
```

```
Sub CmdItaly_Click ()
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
l = 4
MenuPrinc.Show
End Sub
```

```
Sub CmdSpain_Click ()
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
l = 5
MenuPrinc.Show
End Sub
```

```
Sub CmdUK_Click ()
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
```


PRES.FRM - 2

```
1 = 2
MenuPrinc.Show
End Sub
```

```
Sub Command1_Click ()
End
End Sub
```

```
Sub Form_Activate ()
    ecranCour = 1
    Timer1.Enabled = True
    Timer2.Enabled = True
    Timer3.Enabled = True
    Timer4.Enabled = True
    MDI1.Picture1.Visible = False
    MDI1.Histo.Picture = LoadPicture(chem + "telemaqu\dessin\grille0.bmp")
```

```
'mettre les picture de fhisto à invisible
If ecranCour = 24 Then
    cp = 1
    While cp <= cpthist - 1
        fhisto.cmdhisto(cp).Visible = False
        cp = cp + 1
    Wend
End If
cpthist = 1
prechisto = False
```

```
'Remise à zéro du nombre d'impressions totales
nbrimprtot = 0
```

```
End Sub
```

```
Sub Form_Deactivate ()
If boolfin = False Then
    MDI1.MM1.Command = "stop"
    MDI1.MM1.Command = "close"
End If
End Sub
```

```
Sub Form_Unload (cancel As Integer)
cancel = True
End Sub
```

```
Sub Label10_Click ()
If boolfin = False Then
    MDI1.MM1.Command = "stop"
    MDI1.MM1.Command = "close"
End If
DoEvents
boolfin = False
MDI1.MM1.Notify = False
MDI1.MM1.Wait = True
MDI1.MM1.FileName = chem + "telemaqu\son\telemaqu.wav"
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End Sub
```

```
Sub Label8_Click ()
If boolfin = False Then
    MDI1.MM1.Command = "stop"
    MDI1.MM1.Command = "close"
End If
DoEvents
boolfin = False
MDI1.MM1.Notify = False
MDI1.MM1.Wait = True
MDI1.MM1.FileName = chem + "telemaqu\son\telemaqu.wav"
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End Sub
```

```
Sub Label9_Click ()
If boolfin = False Then
    MDI1.MM1.Command = "stop"
```

```
MDI1.MM1.Command = "close"
End If
DoEvents
boolfin = False
MDI1.MM1.Notify = False
MDI1.MM1.Wait = True
MDI1.MM1.FileName = chem + "telemaqu\son\telemaqu.wav"
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End Sub
```

```
Sub Picture1_Click ()
If boolfin = False Then
    MDI1.MM1.Command = "stop"
    MDI1.MM1.Command = "close"
End If
DoEvents
boolfin = False
MDI1.MM1.Notify = False
MDI1.MM1.Wait = True
MDI1.MM1.FileName = chem + "telemaqu\son\adec.wav"
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End Sub
```

```
Sub Picture2_Click ()
If boolfin = False Then
    MDI1.MM1.Command = "stop"
    MDI1.MM1.Command = "close"
End If
DoEvents
boolfin = False
MDI1.MM1.Notify = False
MDI1.MM1.Wait = True
MDI1.MM1.FileName = chem + "telemaqu\son\fundp.wav"
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End Sub
```

```
Sub Picture4_Click ()
If boolfin = False Then
    MDI1.MM1.Command = "stop"
    MDI1.MM1.Command = "close"
End If
DoEvents
boolfin = False
MDI1.MM1.Notify = False
MDI1.MM1.Wait = True
MDI1.MM1.FileName = chem + "telemaqu\son\iecs.wav"
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End Sub
```

```
Sub Timer1_Timer ()
If Picture1.Left < 50 Then
    Picture1.Left = Picture1.Left + c1
    l1 = 1
End If
If Picture1.Left > 10300 Then
    Picture1.Left = Picture1.Left - c1
    l1 = 2
End If
If Picture1.Top < 50 Then
    Picture1.Top = Picture1.Top + c1
    t1 = 1
End If
If Picture1.Top > 6260 Then
    Picture1.Top = Picture1.Top - c1
    t1 = 2
End If
If l1 = 1 Then
    Picture1.Left = Picture1.Left + c1
End If
If l1 = 2 Then
```

PRES.FRM - 4

```
Picture1.Left = Picture1.Left - c1
End If
If t1 = 1 Then
Picture1.Top = Picture1.Top + c1
End If
If t1 = 2 Then
Picture1.Top = Picture1.Top - c1
End If
```

End Sub

Sub Timer2_Timer ()

```
If picture2.Left < 50 Then
picture2.Left = picture2.Left + c2
l2 = 1
End If
If picture2.Left > 10800 Then
picture2.Left = picture2.Left - c2
l2 = 2
End If
If picture2.Top < 50 Then
picture2.Top = picture2.Top + c2
t2 = 1
End If
If picture2.Top > 5760 Then
picture2.Top = picture2.Top - c2
t2 = 2
End If
If l2 = 1 Then
picture2.Left = picture2.Left + c2
End If
If l2 = 2 Then
picture2.Left = picture2.Left - c2
End If
If t2 = 1 Then
picture2.Top = picture2.Top + c2
End If
If t2 = 2 Then
picture2.Top = picture2.Top - c2
End If
```

End Sub

Sub Timer3_Timer ()

```
If l3 = 0 Then
l3 = 1
t3 = 1
End If
If Picture4.Left < 50 Then
Picture4.Left = Picture4.Left + c3
l3 = 1
End If
If Picture4.Left > 10700 Then
Picture4.Left = Picture4.Left - c3
l3 = 2
End If
If Picture4.Top < 50 Then
Picture4.Top = Picture4.Top + c3
t3 = 1
End If
If Picture4.Top > 5960 Then
Picture4.Top = Picture4.Top - c3
t3 = 2
End If
If l3 = 1 Then
Picture4.Left = Picture4.Left + c3
End If
If l3 = 2 Then
Picture4.Left = Picture4.Left - c3
End If
If t3 = 1 Then
Picture4.Top = Picture4.Top + c3
End If
```


PRES.FRM - 5

```
If t3 = 2 Then
Picture4.Top = Picture4.Top - c3
End If
End Sub
```

```
Sub Timer4_Timer ()
timecompt = timecompt + 1
Select Case timecompt
Case 1
    label3.Visible = True
    label7.Visible = False
Case 2
    label4.Visible = True
    label3.Visible = False
Case 3
    label5.Visible = True
    label4.Visible = False
Case 4
    label6.Visible = True
    label5.Visible = fzlse
Case Else
    timecompt = 0
    label7.Visible = True
    label6.Visible = False
End Select
```

End Sub

PRESUNIF.FRM - 1

Const post = ".bmp"

Dim cpt, cpmidi, bstop As Integer

Sub procedfin ()

Dim ij As Integer

If req_univ = "IECS Strasbourg" Then

ij = mciExecute("stop u")

ij = mciExecute("close u")

fentv.Visible = False

fenpourtv.Visible = False

Else

timer1.Enabled = False

timer2.Enabled = False

timer3.Enabled = False

timer4.Enabled = False

labell1.Visible = False

label2.Visible = False

label3.Visible = False

picture2.Visible = False

mm2.Command = "stop"

mm2.Command = "close"

bstop = True

End If

menuunif.Show

End Sub

Sub Command3D1_Click ()

procedfin

End Sub

Sub Form_Activate ()

Dim sql As String

Dim ds_presv As dynaset

Dim ij As Integer

Dim inst As String

'Accès bd pour afficher le logo universitaire

MDI1!Data1.RecordSource = "select LOGO_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "' and NOM_V = '" + req_ville + "' and NOM_U = '" + req_univ + "'"

MDI1!Data1.Refresh

If MDI1!Data1.Recordset("LOGO_U") <> "#NULL#" Then

On Error GoTo errorhandler0

picture3.Picture = LoadPicture(chem + "TELEMAQU\" + Left\$(req_pays, 8) + "\" + Left\$(req_ville, 8) + "\UNIV\" + MDI1!Data1.Recordset("LOGO_U") + ".bmp")

picture3.Visible = True

End If

un:

lbltitre = req_ville

MDI1.Picture1.Visible = False

If req_univ = "IECS Strasbourg" Then

fentv.Visible = True

DoEvents

fenpourtv.Visible = True

inst = "open d:\video\presuniv.avi alias v"

ij = mciExecute(inst)

inst = "window u" + " handle " + Str(fenpourtv.hWnd)

j = mciExecute(inst)

ij = mciExecute("play u")

Else

'Accès BD

sql = "select [N° PRESU], PHOTO PRESU, PHRASEFRA PRESU, PHRASEANG PRESU"

sql = sql + ", PHRASEALL PRESU, PHRASEESP PRESU, PHRASEITA PRESU from PRESUNIV WHERE NOM_P = '"

sql = sql + req_pays + "' and NOM_V = '" + req_ville + "' and NOM_U = '" + req_univ + "' order by [N° PRESU]"

MDI1!Data1.RecordSource = sql

MDI1!Data1.Refresh

Set ds_presu = MDI1!Data1.Recordset

If ds_presu.EOF = False Then

ds_presu.MoveLast

cmpt = ds_presu.RecordCount

ds_presu.MoveFirst

```

'Initialisation du tableau
ReDim tabpresu(cmpt&)
i% = 1
Do While Not (ds_presu.EOF) And i% <= cmpt&
  Select Case 1
    Case 1
      If ds_presu("PHRASEFRA_PRESU") <> "#NULL#" Then
        tabpresu(i%).lblid = ds_presu("PHRASEFRA_PRESU")
      Else
        tabpresu(i%).lblid = ""
      End If
    Case 2
      If ds_presu("PHRASEANG_PRESU") <> "#NULL#" Then
        tabpresu(i%).lblid = ds_presu("PHRASEANG_PRESU")
      Else
        tabpresu(i%).lblid = ""
      End If
    Case 3
      If ds_presu("PHRASEALL_PRESU") <> "#NULL#" Then
        tabpresu(i%).lblid = ds_presu("PHRASEALL_PRESU")
      Else
        tabpresu(i%).lblid = ""
      End If
    Case 4
      If ds_presu("PHRASEITA_PRESU") <> "#NULL#" Then
        tabpresu(i%).lblid = ds_presu("PHRASEITA_PRESU")
      Else
        tabpresu(i%).lblid = ""
      End If
    Case 5
      If ds_presu("PHRASEESP_PRESU") <> "#NULL#" Then
        tabpresu(i%).lblid = ds_presu("PHRASEESP_PRESU")
      Else
        tabpresu(i%).lblid = ""
      End If
  End Select

  If ds_presu("PHOTO_PRESU") <> "#NULL#" Then
    tabpresu(i%).photoid = ds_presu("PHOTO_PRESU")
  Else
    tabpresu(i%).photoid = ""
  End If
  ds_presu.MoveNext
  i% = i% + 1
Loop

```

```

'bande sonore de fond
mm2.Notify = False
mm2.Wait = True
cpmidi = Int(longtabmidi * Rnd + 1)
mm2.FileName = chem + "telemaqu\son\midi\" + tabmidi(cpmidi)
mm2.Command = "open"
mm2.Command = "play"

```

```

EcranCour = 16
initpresunif1
timer1.Interval = 1000
timer2.Interval = 7000
timer3.Interval = 13000
timer4.Interval = 23000
timer1.Enabled = True
timer2.Enabled = True
timer3.Enabled = True
timer4.Enabled = True
cpt = 0
Else
  afficher
  menuunif.Show
End If

```

```

End If
Exit Sub
errorhandler0:
Resume un
End Sub

```



```

Sub Form_Deactivate ()
prechisto = True
picture3.Visible = False
End Sub

```

```

Sub Form_Unload (cancel As Integer)
cancel = True
End Sub

```

```

Sub MM2_Done (NotifyCode As Integer)
'bande sonore de fond
If bstop <> True Then
    mm2.Command = "stop"
    mm2.Command = "close"
    cpmidi = Int(longtabmidi * Rnd + 1)
    mm2.FileName = chem + "telemaqu\son\midi\" + tabmidi(cpmidi)
    mm2.Command = "open"
    mm2.Command = "play"

```

```

End If
End Sub

```

```

Sub Timer1_Timer ()

```

```

    cpt = cpt + 1
    If cpt <= UBound(tabpresu) Then
        label1.Caption = tabpresu(cpt).lblid
        label1.Visible = True
        If tabpresu(cpt).photoid <> "" Then
            On Error GoTo errorhandler
            picture2.Picture = LoadPicture(chem + "telemaqu\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\univ\" + tabpresu(cpt).photoid + post)
            picture2.Visible = True
        End If
    unbis:
    End If
    timer1.Enabled = False
Exit Sub
errorhandler:
Resume unbis
End Sub

```

```

Sub Timer2_Timer ()

```

```

    cpt = cpt + 1
    If cpt <= UBound(tabpresu) Then
        label2.Caption = tabpresu(cpt).lblid
        label2.Visible = True
        If tabpresu(cpt).photoid <> "" Then
            On Error GoTo errorhandler1
            picture2.Picture = LoadPicture(chem + "telemaqu\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\univ\" + tabpresu(cpt).photoid + post)
            picture2.Visible = True
        End If
    unter:
    Else
    procedfin
    End If
    timer2.Enabled = False
Exit Sub
errorhandler1:
Resume unter
End Sub

```

```

Sub Timer3_Timer ()

```

```

    cpt = cpt + 1
    If cpt <= UBound(tabpresu) Then
        label3.Caption = tabpresu(cpt).lblid
        label3.Visible = True
        If tabpresu(cpt).photoid <> "" Then
            On Error GoTo errorhandler2
            picture2.Picture = LoadPicture(chem + "telemaqu\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\univ\" + tabpresu(cpt).photoid + post)
            picture2.Visible = True

```

```
End If
unquatro:
Else
procedfin
End If
timer3.Enabled = False
Exit Sub
errorhandler2:
Resume unquatro
End Sub

Sub Timer4_Timer ()
If cpt < UBound(tabpresu) Then
    label1.Visible = False
    label2.Visible = False
    label3.Visible = False
    picture2.Visible = False
    timer1.Enabled = True
    timer2.Enabled = True
    timer3.Enabled = True
Else
    procedfin
End If
End Sub
```

PRESVILL.FRM - 1

Const post = ".bmp"

Dim cpt, cpmidi, bstop As Integer

Sub procfin ()

Dim ij As Integer

If req_univ = "IECS Strasbourg" Then

ij = mciExecute("stop v")

ij = mciExecute("close v")

fentv.Visible = False

fenpourtv.Visible = False

Else

timer1.Enabled = False

timer2.Enabled = False

timer3.Enabled = False

timer4.Enabled = False

label1.Visible = False

label2.Visible = False

label3.Visible = False

Picture1.Visible = False

mm3.Command = "stop"

mm3.Command = "close"

bstop = True

End If

menuville.Show

End Sub

Sub Cmdstop_Click ()

procfin

End Sub

Sub Form Activate ()

Dim ij As Integer

Dim inst As String

'Accès BD

Dim ds_presv As dynaset

'Accès BD pour y prendre le nom de la ville et le logo de la ville

MDI1!Data1.RecordSource = "select LOGO_V from VILLE WHERE NOM_P = '" + req_pays + "' and NOM_V = '" + req_ville + "'"

MDI1!Data1.Refresh

If MDI1!Data1.Recordset("LOGO_V") <> "#NULL#" Then

On Error GoTo errorHandler0

picture2.Picture = LoadPicture(chem + "TELEMAQU\" + Left\$(req_pays, 8) + "\" + Left\$(req_vill

8) + "\" + MDI1!Data1.Recordset("LOGO_V") + ".bmp")

picture2.Visible = True

End If

un:

lbltitre = req_ville

MDI1.Picture1.Visible = False

If req_univ = "IECS Strasbourg" Then

fentv.Visible = True

DoEvents

fenpourtv.Visible = True

inst = "open d:\video\presuniv.avi alias v"

ij = mciExecute(inst)

inst = "window v" + " handle " + Str(fenpourtv.hWnd)

j = mciExecute(inst)

ij = mciExecute("play v")

Else

MDI1!Data1.RecordSource = "select [N° PRES], PHOTO PRES, PHRASEFRA PRES, PHRASEANG PRES, PHRASEA PRES, PHRASEESP PRES, PHRASEITA PRES from PRESVILLE WHERE NOM_P = '" + req_pays + "' and NOM_V = '" + req_ville + "' order by [N° PRES]"

MDI1!Data1.Refresh

Set ds_presv = MDI1!Data1.Recordset

If ds_presv.EOF = False Then

ds_presv.MoveLast

cmpt& = ds_presv.RecordCount

ds_presv.MoveFirst

'fin accès BD

'Initialisation du tableau

ReDim tabpresv(cmpt&)

i% = 1

Do While Not (ds_presv.EOF) And i% <= cmpt&


```

Select Case 1
Case 1
    If ds_presv("PHRASEFRA_PRES") <> "#NULL#" Then
        tabpresv(i%).lblid = ds_presv("PHRASEFRA_PRES")
    Else
        tabpresv(i%).lblid = ""
    End If
Case 2
    If ds_presv("PHRASEANG_PRES") <> "#NULL#" Then
        tabpresv(i%).lblid = ds_presv("PHRASEANG_PRES")
    Else
        tabpresv(i%).lblid = ""
    End If
Case 3
    If ds_presv("PHRASEALL_PRES") <> "#NULL#" Then
        tabpresv(i%).lblid = ds_presv("PHRASEALL_PRES")
    Else
        tabpresv(i%).lblid = ""
    End If
Case 4
    If ds_presv("PHRASEITA_PRES") <> "#NULL#" Then
        tabpresv(i%).lblid = ds_presv("PHRASEITA_PRES")
    Else
        tabpresv(i%).lblid = ""
    End If
Case 5
    If ds_presv("PHRASEESP_PRES") <> "#NULL#" Then
        tabpresv(i%).lblid = ds_presv("PHRASEESP_PRES")
    Else
        tabpresv(i%).lblid = ""
    End If
End Select

If ds_presv("PHOTO_PRES") <> "#NULL#" Then
    tabpresv(i%).photoid = ds_presv("PHOTO_PRES")
Else
    tabpresv(i%).photoid = ""
End If
ds_presv.MoveNext
i% = i% + 1
Loop

'bande sonore de fond
mm3.Notify = False
mm3.Wait = True
cpmidi = Int(longtabmidi * Rnd + 1)
mm3.FileName = chem + "telemaqu\son\midi\" + tabmidi(cpmidi)
mm3.Command = "open"
mm3.Command = "play"

timer1.Interval = 1000
timer2.Interval = 7000
timer3.Interval = 13000
timer4.Interval = 23000
timer1.Enabled = True
timer2.Enabled = True
timer3.Enabled = True
timer4.Enabled = True
cpt = 0
Else
    afficher
    menuville.Show
End If
End If
Exit Sub
errorhandler0:
Resume un:
End Sub

Sub Form_Deactivate ()
prechisto = True
picture2.Visible = False
End Sub

Sub Form_Unload (cancel As Integer)

```

```
cancel = True
End Sub
```

```
Sub Label1_Click ()
If req_univ = "IECS Strasbourg" Then
    Call setwindowpos(MDI1.hWnd, -1, 0, 0, 0, 0, &H10 Or &H40)
    mm3.Command = "stop"
    mm3.Command = "close"
End If
End Sub
```

```
Sub Label2_Click ()
If req_univ = "IECS Strasbourg" Then
    Call setwindowpos(MDI1.hWnd, -1, 0, 0, 0, 0, &H10 Or &H40)
    mm3.Command = "stop"
    mm3.Command = "close"
End If
End Sub
```

```
Sub Label3_Click ()
If req_univ = "IECS Strasbourg" Then
    Call setwindowpos(MDI1.hWnd, -1, 0, 0, 0, 0, &H10 Or &H40)
    mm3.Command = "stop"
    mm3.Command = "close"
End If
End Sub
```

```
Sub mm3_Done (NotifyCode As Integer)
'bande sonore de fond
If bstop <> True Then
    mm3.Command = "stop"
    mm3.Command = "close"
    cpmidi = Int(longtabmidi * Rnd + 1)
    mm3.FileName = chem + "telemaqu\son\midi\" + tabmidi(cpmidi)
    mm3.Command = "open"
    mm3.Command = "play"
End If
End Sub
```

```
Sub Picture1_Click ()
If req_univ = "IECS Strasbourg" Then
    Call setwindowpos(MDI1.hWnd, -1, 0, 0, 0, 0, &H10 Or &H40)
    mm3.Command = "stop"
    mm3.Command = "close"
End If
End Sub
```

```
Sub Timer1_Timer ()
cpt = cpt + 1
If cpt <= UBound(tabpresv) Then
    label1.Caption = tabpresv(cpt).lblid
    label1.Visible = True
    If tabpresv(cpt).photoid <> "" Then
        On Error GoTo errorhandler
        Picture1.Picture = LoadPicture(chem + "telemaqu\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\" + ville\" + tabpresv(cpt).photoid + post)
        Picture1.Visible = True
    End If
unbis:
End If
timer1.Enabled = False
Exit Sub
errorhandler:
Resume unbis
End Sub
```

```
Sub Timer2_Timer ()
cpt = cpt + 1
If cpt <= UBound(tabpresv) Then
    label2.Caption = tabpresv(cpt).lblid
    label2.Visible = True
    If tabpresv(cpt).photoid <> "" Then
        On Error GoTo errorhandler1
        Picture1.Picture = LoadPicture(chem + "telemaqu\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\" + ville\" + tabpresv(cpt).photoid + post)
    End If
End Sub
```

```
        Picture1.Visible = True
    End If
deuxbis:
Else
    procfin
End If
timer2.Enabled = False
Exit Sub
errorhandler1:
Resume deuxbis
End Sub

Sub Timer3_Timer ()
    cpt = cpt + 1
    If cpt <= UBound(tabpresv) Then
        label3.Caption = tabpresv(cpt).lblid
        label3.Visible = True
        If tabpresv(cpt).photoid <> "" Then
            On Error GoTo errorhandler3
            Picture1.Picture = LoadPicture(chem + "telemaqu\" + Left$(req_pays, 8) + "\" + Left$(req_
lle, 8) + "\ville\" + tabpresv(cpt).photoid + post)
            Picture1.Visible = True
        End If
    End If
troisbis:
Else
    procfin
End If
timer3.Enabled = False
Exit Sub
errorhandler3:
Resume troisbis
End Sub

Sub Timer4_Timer ()
    If cpt < UBound(tabpresv) Then
        label1.Visible = False
        label2.Visible = False
        label3.Visible = False
        Picture1.Visible = False
        timer1.Enabled = True
        timer2.Enabled = True
        timer3.Enabled = True
    Else
        procfin
    End If
End Sub
```


STAT1.FRM - 1

Dim cptgr As Integer

Sub Form_Activate ()

If nonfirst = True Then

graphel.Visible = False

EcranCour = 18

lbltitre.Caption = req_ville

NOM V=" + req_ville + " and NOM_U=" + req_univ + ""

MDI1!Data1.Refresh

Set ds_unif = MDI1!Data1.Recordset

If ds_unif("LOGO_U") <> "#NULL#" Then

On Error GoTo errorhandler

picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left\$(req_pays, 8) + "\" + Left\$(req_

lle, 8) + "\" + Left\$(req_

picture1.Visible = True

End If

un:

comptgraf = 1

initstat1

If comptgraf < UBound(tabg) Then

suiv.Visible = True

End If

prec.Visible = False

prochisto

Else

End If

Exit Sub

errorhandler:

Resume un

End Sub

Sub Form_Deactivate ()

picture1.Visible = False

End Sub

Sub Form_Unload (cancel As Integer)

cancel = True

End Sub

Sub Picture3_Click ()

guignols

End Sub

Sub prec_Click ()

If boolaide = False Then

comptgraf = comptgraf - 1

initstatlbis

initstatlcl

If comptgraf = 1 Then

prec.Visible = False

End If

suiv.Visible = True

Else

If boolfin = False Then

MDI1.MM1.Command = "stop"

MDI1.MM1.Command = "close"

End If

DoEvents

boolfin = False

MDI1.MM1.FileName = tabhlp2(1)

MDI1.MM1.Command = "open"

MDI1.MM1.Command = "play"

End If

End Sub

Sub suiv_Click ()

If boolaide = False Then

comptgraf = comptgraf + 1

initstatlbis

initstatlcl

If comptgraf = UBound(tabg) Then

suiv.Visible = False

STAT1.FRM - 2

```
End If
prec.Visible = True
Else
  If boolfin = False Then
    MDI1.MM1.Command = "stop"
    MDI1.MM1.Command = "close"
  End If
  DoEvents
  boolfin = False
  MDI1.MM1.FileName = tabhlp2(1)
  MDI1.MM1.Command = "open"
  MDI1.MM1.Command = "play"
End If
End Sub
```

DECLARE.BAS - 1

Option Base 1

'Pour garder VB à l'avant plan.

Declare Sub setwindowpos Lib "User" (ByVal hWnd As Integer, ByVal hWndInsertAfter As Integer, ByVal X As Integer, ByVal Y As Integer, ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags As Integer)

'Pour exécuter la vidéo

Declare Function mciExecute Lib "mmsystem" (ByVal lpstrCommand As String) As Integer

'Variable utilisée pour localiser d'où viennent les
'erreurs trappées

Global arret As String

'Variable pour indiquer le nom du drice

Global chem As String

'Voyelle pour la langue espagnole ñáéíóú

'Permet de savoir lors de initbibliobis s'il y a des infos à afficher

'(pour ne pas afficher derrière la msgbox, des infos de la ville précédente

Global pasdinfo As Integer

'Constante nombre de langues

Global Const nbl = 5

'Contient le nombre de items pour le ds_orga

Global nbrorga As Integer

'Pour gérer le changement de langue dans Orga

Global chl As Integer

,

Global cporga As Integer

'Pour mettre les noms et les traductions des postes pour l'organisation

Global taborgan() As String

Global taborgafct() As String

'Booléen pour signaler si je viens de l'écran cours au

'niveau des groupes ou au niveau des cours

'Initialisé dans cursus deactivate et dans anigroupe()

Global groupe As Integer

'variable globale servant à identifier la langue courante

'1=français 2=anglais 3=allemand 4=italien 5=espagnol

Global l As Integer

'Indique le nombre d'éléments à mettre dans la liste

'déroulante

Global bornesup As Integer

'Trois tableaux pour vider le ds_cours

Global tabintit() As String

Global tabh() As String

Global tabp() As String

'Dynaset pour informations sur les cours

Global ds_cours As dynaset

'Tableau temporaire pour faire correspondre les index des noms de groupe dans les boutons avec un
tableau

Global Tabtempgrp(9) As String

'var contenant le nom du groupe choisi

Global groupechoisi As String

'Pour la form aubiblio

Global cp As Integer

'Contient le nombre d'années dans une fac

Global how As Integer

'tableau de booléens = vrai s'il n'y a qu'une date pour les dates à retenir

DECLARE.BAS - 2

Global unedate() As Integer

'Contient le nombre d'enregistrement pour un type d'infos donnés
Global nbrinfos As Integer

'Contient le nombre de groupe pour une année
Global nbrgroupe As Integer

'Contient les différents groupes
Global tabgroupe() As String

'Variable pour faire les requetes sur la ville du pays
Global req_ville As String

'Variable pour les req sur le nom de la fac
Global req_fac As String

'Variable pour faire les requêtes sur une université d'une ville, d'un pays
Global req_univ As String

'Variable pour faire les requêtes sur les infos
Global req_info As String

'Variable pour faire les requetes sur le pays
Global req_pays As String

'Booléen vrai si on vient d'appuyer sur le
'changement de langue
Global chgtl As Integer

'Tableau pour la traduction des noms de pays au niveau
'de la forme unif-part
Global tabenspays(0 To 30, nbl) As String

'Tableau comprenant
Global tabpaysdans() As Integer

'Va indiquer l'indice du tableau tabenspays, pour la
'traduction des pays
Global pays_ok As Integer

'Dynaset qui donne tous les pays repris dans la BD
Global ds As dynaset

'Dynaset qui donne la présentation d'une unif
Global ds_presu As dynaset

'Dynaset qui donne la présentation d'une ville
Global ds_presv As dynaset

'Dynaset qui donne toutes les villes et leurs numeros dans un pays
Global ds_Ville As dynaset

'Dynaset qui donne toutes les universités dans une ville
Global ds_unif As dynaset

'Variable contenant le nombre de pays dans la bbase
'de données
Global nbrpays As Integer

'Constante nombre de formes
Global Const nbf = 40

'booléen pour determiner si l'écran précédent est
'l'historique
Global prechisto As Integer

'tableau pour gerer l'historique
Type recinfo
 numero As Integer
 country As String
 city As String
 univer As String
 depart As String

DECLARE.BAS - 3

 langue As Integer
End Type

Global tabhisto(9) As recinfo

'Compteur pour voyager dans ce tableau
Global Sortietab As Integer

'Variable globale servant à identifier l'écran courant
Global ecrancour As Integer

'variable servant à donner une valeur aux labels secondaires
Global sstitre(nbl) As String

'Variable servant à contenir le pays courant
Global pays As String

'tableaux de l'écran MenuPrinc
Global tabUnifPart(nbl) As String

'tableau de l'écran MenuUnifPart (pour le sous-titre)
Global tabchoixregion(nbl) As String
Global tabchoixpays(nbl) As String
Global tabpc(nbl) As String

'tableaux de l'écran menuVille
Global tabpresville(nbl), tabinfunif(nbl), tabinfprat(nbl) As String

'Tableau pour les noms d'années en 5 langues.(écran cursus)
Global tabannee(5, nbl) As String

'tableaux de l'écran MenuUnif
Global tabpresunif(nbl), tabchiffres(nbl), tabdepiecs(nbl) As String

'tableaux de l'écran MenuIecs
Global tabcurs(nbl), taborga(nbl), tabcondad(nbl)

'tableaux de l'écran MenuInfos
Global tabadrut(nbl), tabviequot(nbl), tabplan(nbl), tabcalend(nbl) As String

'tableaux de l'écran MnuAUt1
Global tabAUAssocEt(nbl), tabAUUnif(nbl), tabAUSecr(nbl), tabAUCentre(nbl), tabAUCentre2(nbl), tabUBiblio(nbl) As String

'tableaux de l'écran MenuVieQuot
Global tabVQLog(nbl), tabVQRestaus(nbl), tabVQBanques(nbl), tabVQPostes(nbl), tabVQCinTh(nbl), tabVQSport(nbl) As String

'tableau de l'écran introville
Global tabintroville(nbl) As String

'Tableau de drapeaux
Global tabflag(nbl) As String

'tableau pour l'icône imprimante
Global tabimpr(2) As String

'Tableaux reprenant la suite des formes et leur prédécesseur
Global tabidform(nbf) As form
Global tabprecform(nbf) As Integer

'Variables concernant le système d'aide
Global boolaide As Integer
Global boolfin As Integer

'Tableau pour les sons
Global tabhlp1(nbl) As String
Global tabhlp2(nbl) As String
Global tabhist(nbl) As String
Global tablang(nbl) As String
Global tabmenu(nbl) As String
Global tabprnt(nbl) As String
Global tabquit(nbl) As String
Global tabreto(nbl) As String

DECLARE.BAS - 4

Global tabunif(nbl) As String
Global tabpl(nbl) As String

'Tableau pour les graphiques de stat
Global tabstat(2) As String

'Record de deux strings pour les présentations
Type rec
 lblid As String
 photoid As String
End Type

'Déclaration d'une variable de ce type
Global tabpresu() As rec

Global tabpresv() As rec

'compteur pour le logo historique
Global cpthist As Integer

'Booléen pour tester si on est dans la première série de Form Activate
Global Nonfirst As Integer

'Pour mettre les noms en cinq langues des infos pratiques
Global tabinitbiblionom() As String

'Pour mettre les remarques (adresses) en cinq langues des infos pratiques
Global tabinitbibliorem() As String

'Pour mettre les dates de début et de fin des infos pratiques, de type dates à retenir
Global tabinitbibliodated() As String
Global tabinitbibliodatef() As String

'Pour mettre les renseignements, si type_inf <> dates à retenir
Global tabinitbibliorens() As String

'Tableaux pour InitBiblio , remplacé par ce qui est ci-dessus
Type unite
 nom As String
 adr As String
 descr As String
End Type

Global tabinitbiblio() As unite

'Tableau pour les sous-titres
Global tablbl(nbl, nbf) As String

'Pour savoir quelle année traiter après cursus
Global cu As Integer

'Pour savoir quelle groupe traiter dans cours
Global cubis As Integer

'Tableau pour la procédure afficher (dans traitement.bas)
'Gère l'affichage d'un message à l'écran
Global tabmess(nbl) As String
Global tabtitremess(nbl) As String
Global tabimpression(nbl) As String
Global tabtitreimpr(nbl) As String

'Dynasets pour les graphiques (écran stat1)
Dim ds_graf As dynaset
Dim ds_chiffres As dynaset

'Tableau pour les graphiques (écran stat1)
'initstatlbis
Type tchiffres
 nomch(nbl) As String
 ordo As String
End Type
Global tabchiffre() As tchiffres

'initstat1
Type tgraf

DECLARE.BAS - 5

```
    tabnomgraf(nbl) As String
    typ As String
    absc As String
End Type
Global tabg() As tgraf

'initstatlbis
Type tordo
    valeurs(50) As Integer
End Type
'pour simuler un tableau de tableau.
'Pq 50 ? car on ne peut pas déclarer
'un tableau dynamique comme champ d'un record

'Donne le numéro du graphe à afficher
Global comptgraf As Integer

'pour mettre nom_p, nom_V, nom_U
Global ds_concat As dynaset

'Tableau pour le changement d'université
Type paviun
    pa As String
    vi As String
    un As String
End Type

Global Tabchgtuniv() As paviun

Type total
    pay As Integer
    viluniv As String
    indice As Integer
End Type

Global tabconcat() As total

'Var contenant l'indice de référence vers le
'tableau tabchgtuniv
Global choixuniv As Integer

'Contient les traductions du titre pour le changement
'd'université
Global tabtitrechgtu(nbl) As String

'Contient les traductions du titre pour l'historique
Global tabtitrehisto(nbl) As String

'tableau des fichiers MIDI
Global tabmidi(30) As String
Global longtabmidi As Integer

'tableau des fichiers WAVE
Global tabwave(40) As String
Global longtabwave As Integer

'Pour l'initialisation des tableaux tabmidi et tabwave
Dim repert As String

Sub chargement ()
    l = 1
    'Load AUBiblio
    'Load cours
    'Load cursus
    'Load fhisto
    'Load introville
    'Load MenuIECS
    'Load MenuInfos
    'Load MenuUnifPart
    'Load MenuPays
    'Load menuprinc
    'Load MenuUnif
    'Load MenuVieQuot
    'Load menuville
    'Load MnuAUtl
    'Load orga
    'Load plans
```

DECLARE.BAS - 6

```
'Load Pres
'Load presunif1
'Load presville
'Load Stat1
'Load FBlanche
,
```

End Sub

Sub InitBiblio ()

'Déclaration des variables

Dim sql As String 'Contient la requête SQL

Dim ds_info As dynaset 'Contient le résultat de la requête SQL

Dim reqtemp As String

req_info = sstitre(1)

AUBiblio.lblsstitre = sstitre(1)

If req_info = "A qui s'adresser" Then req_info = "A qui s'adresser"

If req_info = "Associations d'étudiants" Then req_info = "Associations d'étudiants"

If req_info = "Dates à retenir" Then

reqtemp = "select NOMFRA_INF, NOMANG_INF, NOMALL_INF, NOMESP_INF, NOMITA_INF,"

reqtemp = reqtemp + " DATED, DATEF, REMARQUEFRA_INF, REMARQUEANG_INF,"

reqtemp = reqtemp + "REMARQUEALL_INF, REMARQUEITA_INF, REMARQUEESP_INF from INFOPRAT WHERE NO

P = ""

sql = reqtemp + req_pays + "" and NOM_V= "" + req_ville + "" and TYPE_INF = "" + req_info + ""

order by DATED"

Else

reqtemp = "select NOMFRA_INF, NOMANG_INF, NOMALL_INF, NOMESP_INF, NOMITA_INF,"

reqtemp = reqtemp + " RENSEIGNEMENT, REMARQUEFRA_INF, REMARQUEANG_INF,"

reqtemp = reqtemp + "REMARQUEALL_INF, REMARQUEITA_INF, REMARQUEESP_INF from INFOPRAT WHERE NO

P = ""

sql = reqtemp + req_pays + "" and NOM_V= "" + req_ville + "" and TYPE_INF = "" + req_info + ""

order by NOMFRA_INF"

End If

MDI1!Data1.RecordSource = sql

MDI1!Data1.Refresh

Set ds_info = MDI1!Data1.Recordset

If ds_info.EOF = False Then

pasdinfo = False

ds_info.MoveLast

cmpt& = ds_info.RecordCount

ds_info.MoveFirst

nbrinfos = cmpt&

ReDim tabinitbiblionom(cmpt&, 5)

ReDim tabinitbibliorem(cmpt&, 5)

ReDim tabinitbibliodated(cmpt&)

ReDim tabinitbibliodatef(cmpt&)

ReDim tabinitbibliorens(cmpt&)

ReDim unedate(cmpt&)

'Initialisation du tableau à False

For k% = 1 To cmpt&

unedate(k%) = False

Next k%

i% = 1

Do While Not ds_info.EOF

If ds_info("NOMFRA_INF") <> "#NULL#" Then

tabinitbiblionom(i%, 1) = ds_info("NOMFRA_INF")

Else

tabinitbiblionom(i%, 1) = ""

End If

If ds_info("NOMANG_INF") <> "#NULL#" Then

tabinitbiblionom(i%, 2) = ds_info("NOMANG_INF")

Else

tabinitbiblionom(i%, 2) = ""

End If

If ds_info("NOMALL_INF") <> "#NULL#" Then

tabinitbiblionom(i%, 3) = ds_info("NOMALL_INF")

Else

tabinitbiblionom(i%, 3) = ""

End If

```

    If ds_info("NOMITA_INF") <> "#NULL#" Then
        tabinitbiblionom(i%, 4) = ds_info("NOMITA_INF")
    Else
        tabinitbiblionom(i%, 4) = ""
    End If
    If ds_info("NOMESP_INF") <> "#NULL#" Then
        tabinitbiblionom(i%, 5) = ds_info("NOMESP_INF")
    Else
        tabinitbiblionom(i%, 5) = ""
    End If
    If req_info = "Dates à retenir" Then
        If ds_info("DATED") <> "#NULL#" Then
            tabinitbibliodated(i%) = Format(ds_info("DATED"), "DD/MM")
        End If
        If ds_info("DATEF") <> "#NULL#" Then
            tabinitbibliodatef(i%) = Format(ds_info("DATEF"), "DD/MM")
        Else
            unedate(i%) = True
        End If
    Else
        If ds_info("RENSEIGNEMENT") <> "#NULL#" Then
            tabinitbibliorens(i%) = ds_info("RENSEIGNEMENT")
        End If
    End If

    If ds_info("REMARQUEFRA_INF") <> "#NULL#" Then
        tabinitbibliorem(i%, 1) = ds_info("REMARQUEFRA_INF")
    Else
        tabinitbibliorem(i%, 1) = ""
    End If
    If ds_info("REMARQUEANG_INF") <> "#NULL#" Then
        tabinitbibliorem(i%, 2) = ds_info("REMARQUEANG_INF")
    Else
        tabinitbibliorem(i%, 2) = ""
    End If
    If ds_info("REMARQUEALL_INF") <> "#NULL#" Then
        tabinitbibliorem(i%, 3) = ds_info("REMARQUEALL_INF")
    Else
        tabinitbibliorem(i%, 3) = ""
    End If
    If ds_info("REMARQUEITA_INF") <> "#NULL#" Then
        tabinitbibliorem(i%, 4) = ds_info("REMARQUEITA_INF")
    Else
        tabinitbibliorem(i%, 4) = ""
    End If
    If ds_info("REMARQUEESP_INF") <> "#NULL#" Then
        tabinitbibliorem(i%, 5) = ds_info("REMARQUEESP_INF")
    Else
        tabinitbibliorem(i%, 5) = ""
    End If

    ds_info.MoveNext
    i% = i% + 1
Loop
Else
    pasdinfo = True
    afficher
    tabidform(tabprecform(ecrancour)).Show
End If
End Sub

Sub initbibliobis ()
    If pasdinfo = False Then
        Const nbre = 4
        AUBiblio.lblsstitre = sstitre(1)
        compteur = 0
        While (cp <= nbrinfos) And (compteur < nbre)
            AUBiblio!lblnom(compteur).Caption = tabinitbiblionom(cp, 1)
            If req_info = "Dates à retenir" Then
                If unedate(cp) = False Then
                    AUBiblio!lbladr(compteur).Caption = tabinitbibliodated(cp) + " - " + tabinitbibl
                Else
                    AUBiblio!lbladr(compteur).Caption = tabinitbibliodatef(cp)
                End If
            Else
                AUBiblio!lbladr(compteur).Caption = tabinitbibliorens(cp)
            End If
            compteur = compteur + 1
        End While
    End If
End Sub

```


DECLARE.BAS - 8

```
Else
    AUBiblio!lbladr(compteur).Caption = tabinitbibliorens(cp)
End If
AUBiblio!lbldescr(compteur).Caption = tabinitbibliorem(cp, 1)

AUBiblio!lblnom(compteur).Visible = True
AUBiblio!lbladr(compteur).Visible = True
AUBiblio!lbldescr(compteur).Visible = True
cp = cp + 1
compteur = compteur + 1
Wend

If cp <= nbrinfos Then
    AUBiblio!suiv.Visible = True
End If
cp = cp - compteur
End If

End Sub

Sub InitCalend ()
    Calend.lblsstitre = tablbl(1, ecrancour)
End Sub

Sub initchgtuniv ()
    ChgtUniv!lbltitre = tabtitrechgtu(1)
End Sub

Sub initchgtunivbis ()
    ChgtUniv!lbltitre = tabtitrechgtu(1)

If UBound(tabconcat) > 11 Then
    For p% = 0 To 9
        If tabconcat(ChgtUniv!ascen.Value + p% + 1).indice = 9999 Then
            ChgtUniv!Labell(p%).Caption = tabenspays(tabconcat(ChgtUniv!ascen.Value + p% + 1).pay
1)
            ChgtUniv!Labell(p%).Tag = "9999"
        Else
            ChgtUniv!Labell(p%).Caption = tabconcat(ChgtUniv!ascen.Value + p% + 1).viluniv
            ChgtUniv!Labell(p%).Tag = Str(tabconcat(ChgtUniv!ascen.Value + p% + 1).indice)
        End If
    Next p%
Else
    i% = 1
    While (i% <= UBound(tabconcat) - 1)
        If tabconcat(i%).indice = 9999 Then
            ChgtUniv!Labell(i% - 1).Caption = tabenspays(tabconcat(i%).pay, 1)
            ChgtUniv!Labell(i% - 1).Tag = "9999"
            ChgtUniv!Labell(i% - 1).ForeColor = &HB14B07
        Else
            ChgtUniv!Labell(i% - 1).Caption = tabconcat(i%).viluniv
            ChgtUniv!Labell(i% - 1).Tag = Str(tabconcat(i%).indice)
        End If
        ChgtUniv!Labell(i% - 1).Visible = True
        i% = i% + 1
    Wend
End If
End Sub

Sub InitCondAdm ()
    CondAdm.lblsstitre.Caption = tablbl(1, ecrancour)
End Sub

Sub initcours ()
    cours.lblsstitre.Caption = tabanee(cu, 1)
    If groupe = True Then
        Select Case nbrgroupe
            Case 1
                cours!Anigroupe(5).Visible = True
                cours!Anigroupe(5).Caption = tabgroupe(1, 1)
            Case 2
```

```
cours!Anigroupe(2).Caption = tabgroupe(1, 1)
cours!Anigroupe(2).Visible = True
cours!Anigroupe(5).Caption = tabgroupe(2, 1)
cours!Anigroupe(5).Visible = True
```

Case 3

```
cours!Anigroupe(4).Caption = tabgroupe(1, 1)
cours!Anigroupe(4).Visible = True
cours!Anigroupe(5).Caption = tabgroupe(2, 1)
cours!Anigroupe(5).Visible = True
cours!Anigroupe(6).Caption = tabgroupe(3, 1)
cours!Anigroupe(6).Visible = True
```

Case 4

```
cours!Anigroupe(2).Caption = tabgroupe(1, 1)
cours!Anigroupe(2).Visible = True
cours!Anigroupe(4).Caption = tabgroupe(2, 1)
cours!Anigroupe(4).Visible = True
cours!Anigroupe(6).Caption = tabgroupe(3, 1)
cours!Anigroupe(6).Visible = True
cours!Anigroupe(8).Caption = tabgroupe(4, 1)
cours!Anigroupe(8).Visible = True
```

Case 5

```
cours!Anigroupe(1).Caption = tabgroupe(1, 1)
cours!Anigroupe(1).Visible = True
cours!Anigroupe(3).Caption = tabgroupe(2, 1)
cours!Anigroupe(3).Visible = True
cours!Anigroupe(5).Caption = tabgroupe(3, 1)
cours!Anigroupe(5).Visible = True
cours!Anigroupe(7).Caption = tabgroupe(4, 1)
cours!Anigroupe(7).Visible = True
cours!Anigroupe(9).Caption = tabgroupe(5, 1)
cours!Anigroupe(9).Visible = True
```

Case 6

```
For i% = 1 To nbrgroupe
    cours!Anigroupe(i%).Caption = tabgroupe(i%, 1)
    cours!Anigroupe(i%).Visible = True
Next i%
```

Case 7

```
cours!Anigroupe(1).Caption = tabgroupe(1, 1)
cours!Anigroupe(1).Visible = True
cours!Anigroupe(2).Caption = tabgroupe(2, 1)
cours!Anigroupe(2).Visible = True
cours!Anigroupe(3).Caption = tabgroupe(3, 1)
cours!Anigroupe(3).Visible = True
cours!Anigroupe(5).Caption = tabgroupe(4, 1)
cours!Anigroupe(5).Visible = True
cours!Anigroupe(7).Caption = tabgroupe(5, 1)
cours!Anigroupe(7).Visible = True
cours!Anigroupe(8).Caption = tabgroupe(6, 1)
cours!Anigroupe(8).Visible = True
cours!Anigroupe(9).Caption = tabgroupe(7, 1)
cours!Anigroupe(9).Visible = True
```

Case 8

```
cours!Anigroupe(1).Caption = tabgroupe(1, 1)
cours!Anigroupe(1).Visible = True
cours!Anigroupe(2).Caption = tabgroupe(2, 1)
cours!Anigroupe(2).Visible = True
cours!Anigroupe(3).Caption = tabgroupe(3, 1)
cours!Anigroupe(3).Visible = True
cours!Anigroupe(4).Caption = tabgroupe(4, 1)
cours!Anigroupe(4).Visible = True
cours!Anigroupe(6).Caption = tabgroupe(5, 1)
cours!Anigroupe(6).Visible = True
cours!Anigroupe(7).Caption = tabgroupe(6, 1)
cours!Anigroupe(7).Visible = True
cours!Anigroupe(8).Caption = tabgroupe(7, 1)
cours!Anigroupe(8).Visible = True
cours!Anigroupe(9).Caption = tabgroupe(8, 1)
cours!Anigroupe(9).Visible = True
```

Case 9

```
For i% = 1 To nbrgroupe
```

DECLARE.BAS - 10

```
                cours!Anigroupe(i%).Caption = tabgroupe(i%, 1)
                cours!Anigroupe(i%).Visible = True
            Next i%
        End Select
    Else
        Select Case bornesup
        Case 1
            cours!lblintit(1) = tabintit(cours!scroll.Value + 1, 1)
            cours!lblprof(1) = tabp(cours!scroll.Value + 1)
            cours!lblheures(1) = tabh(cours!scroll.Value + 1)

        Case 2
            cours!lblintit(1) = tabintit(cours!scroll.Value + 1, 1)
            cours!lblprof(1) = tabp(cours!scroll.Value + 1)
            cours!lblheures(1) = tabh(cours!scroll.Value + 1)
            cours!lblintit(2) = tabintit(cours!scroll.Value + 2, 1)
            cours!lblprof(2) = tabp(cours!scroll.Value + 2)
            cours!lblheures(2) = tabh(cours!scroll.Value + 2)

        Case Else
            cours!lblintit(1) = tabintit(cours!scroll.Value + 1, 1)
            cours!lblprof(1) = tabp(cours!scroll.Value + 1)
            cours!lblheures(1) = tabh(cours!scroll.Value + 1)
            cours!lblintit(2) = tabintit(cours!scroll.Value + 2, 1)
            cours!lblprof(2) = tabp(cours!scroll.Value + 2)
            cours!lblheures(2) = tabh(cours!scroll.Value + 2)
            cours!lblintit(3) = tabintit(cours!scroll.Value + 3, 1)
            cours!lblprof(3) = tabp(cours!scroll.Value + 3)
            cours!lblheures(3) = tabh(cours!scroll.Value + 3)

        End Select
    End If
End Sub

Sub InitCursus ()
cursus.lblsstitre.Caption = tablbl(1, ecrancour)
    For i% = 1 To how
        cursus!CmdAnnée(i%).Caption = tabannee(i%, 1)
    Next i%
End Sub

Sub initfhisto ()
fhisto!lblsstitre = tabtitrehisto(1)
End Sub

Sub initialisation ()
chem = Left$(CurDir, 3)
'l'écran précédent n'est pas l'historique
prechisto = False

'initialisation du compteur pour l'image de l'historique
cpthist = 1

'Compteur dans le tableau tabhisto
Sortietab = 1

'titre de l'écran Historique
tabtitrehisto(1) = "Appuyez sur l'écran que vous voulez revoir"
tabtitrehisto(2) = "histoang"
tabtitrehisto(3) = "histoall"
tabtitrehisto(4) = "histoita"
tabtitrehisto(5) = "histoesp"

'titre de l'écran Chgtuniv
tabtitrechgtu(1) = "Changement d'université"
tabtitrechgtu(2) = "University switch"
tabtitrechgtu(3) = "chgtall"
tabtitrechgtu(4) = "chgtita"
tabtitrechgtu(5) = "chgtesp"

'Tableau pour les années de la form cursus
tabannee(1, 1) = "Première année"
```


DECLARE.BAS - 11

```
tabannee(1, 2) = "First year"
tabannee(1, 3) = ""
tabannee(1, 4) = ""
tabannee(1, 5) = ""
tabannee(2, 1) = "Deuxième année"
tabannee(2, 2) = "Second year"
tabannee(2, 3) = ""
tabannee(2, 4) = ""
tabannee(2, 5) = ""
tabannee(3, 1) = "Troisième année"
tabannee(3, 2) = "Third year"
tabannee(3, 3) = ""
tabannee(3, 4) = ""
tabannee(3, 5) = ""
tabannee(4, 1) = "Quatrième année"
tabannee(4, 2) = "Fourth year"
tabannee(4, 3) = ""
tabannee(4, 4) = ""
tabannee(4, 5) = ""
tabannee(5, 1) = "Cinquième année"
tabannee(5, 2) = "Fifth year"
tabannee(5, 3) = ""
tabannee(5, 4) = ""
tabannee(5, 5) = ""
'Pour éviter le plantage lors du loadage des form au tout début
pays_ok = 0
```

'Tableau pour la traduction des noms de pays

'All=1, Pays_Bas=2, Belg=3, Dane=4,finlande=5, suede=6

'G-B =7, Irlande=8, canada=9,USA=10, Grèce=11,Autriche=12

'Suisse=13, Italie=14, Espagne=15,

'rmq: il faut laisser le 0 à blanc pour permettre des initialisations correctes

```
tabenspays(0, 1) = ""
tabenspays(0, 2) = ""
tabenspays(0, 3) = ""
tabenspays(0, 4) = ""
tabenspays(0, 5) = ""
```

```
tabenspays(1, 1) = "Allemagne"
tabenspays(1, 2) = "Germany"
tabenspays(1, 3) = ""
tabenspays(1, 4) = ""
tabenspays(1, 5) = "Alemania"
```

```
tabenspays(2, 1) = "Pays-Bas"
tabenspays(2, 2) = "Netherland"
tabenspays(2, 3) = ""
tabenspays(2, 4) = ""
tabenspays(2, 5) = "Holanda"
```

```
tabenspays(3, 1) = "Belgique-Luxembourg"
tabenspays(3, 2) = "Belgium-Luxembourg"
tabenspays(3, 3) = ""
tabenspays(3, 4) = ""
tabenspays(3, 5) = "Bélgica"
```

```
tabenspays(4, 1) = "Danemark"
tabenspays(4, 2) = "Danemark"
tabenspays(4, 3) = ""
tabenspays(4, 4) = ""
tabenspays(4, 5) = "Dinamarca"
```

```
tabenspays(5, 1) = "Finlande"
tabenspays(5, 2) = "Finland"
tabenspays(5, 3) = ""
tabenspays(5, 4) = ""
tabenspays(5, 5) = "Finlandia"
```

```
tabenspays(6, 1) = "Suède"
tabenspays(6, 2) = "Sweden"
tabenspays(6, 3) = ""
tabenspays(6, 4) = ""
tabenspays(6, 5) = "Suecia"
```

DECLARE.BAS - 12

```
tabenspays(7, 1) = "Grande-Bretagne"  
tabenspays(7, 2) = "Great Britain"  
tabenspays(7, 3) = ""  
tabenspays(7, 4) = ""  
tabenspays(7, 5) = "Gran Bretaña"
```

```
tabenspays(8, 1) = "Irlande"  
tabenspays(8, 2) = "Ireland"  
tabenspays(8, 3) = ""  
tabenspays(8, 4) = ""  
tabenspays(8, 5) = "Irlanda"
```

```
tabenspays(9, 1) = "Reste du monde"  
tabenspays(9, 2) = "Rest of the world"  
tabenspays(9, 3) = ""  
tabenspays(9, 4) = ""  
tabenspays(9, 5) = ""
```

```
tabenspays(10, 1) = "France"  
tabenspays(10, 2) = "France"  
tabenspays(10, 3) = ""  
tabenspays(10, 4) = ""  
tabenspays(10, 5) = ""
```

```
tabenspays(11, 1) = "Grèce"  
tabenspays(11, 2) = "Greece"  
tabenspays(11, 3) = ""  
tabenspays(11, 4) = ""  
tabenspays(11, 5) = "Grecia"
```

```
tabenspays(12, 1) = "Autriche"  
tabenspays(12, 2) = "Austria"  
tabenspays(12, 3) = ""  
tabenspays(12, 4) = ""  
tabenspays(12, 5) = "Austria"
```

```
tabenspays(13, 1) = "Suisse"  
tabenspays(13, 2) = "Switzerland"  
tabenspays(13, 3) = ""  
tabenspays(13, 4) = ""  
tabenspays(13, 5) = "Suiza"
```

```
tabenspays(14, 1) = "Italie"  
tabenspays(14, 2) = "Italy"  
tabenspays(14, 3) = ""  
tabenspays(14, 4) = ""  
tabenspays(14, 5) = "Italia"
```

```
tabenspays(15, 1) = "Espagne"  
tabenspays(15, 2) = "Spain"  
tabenspays(15, 3) = ""  
tabenspays(15, 4) = ""  
tabenspays(15, 5) = "España"
```

```
tabenspays(16, 1) = "Islande"  
tabenspays(16, 2) = "Island"  
tabenspays(16, 3) = ""  
tabenspays(16, 4) = ""  
tabenspays(16, 5) = ""
```

```
tabenspays(17, 1) = "Norvège"  
tabenspays(17, 2) = "Norwegia"  
tabenspays(17, 3) = ""  
tabenspays(17, 4) = ""  
tabenspays(17, 5) = ""
```

```
tabenspays(18, 1) = "Pologne"  
tabenspays(18, 2) = "Poland"  
tabenspays(18, 3) = ""  
tabenspays(18, 4) = ""  
tabenspays(18, 5) = ""
```

```
tabenspays(19, 1) = "Portugal"  
tabenspays(19, 2) = "Portugal"
```

DECLARE.BAS - 13

```
tabenspays(19, 3) = ""
tabenspays(19, 4) = ""
tabenspays(19, 5) = ""
```

```
tabenspays(20, 1) = "Tchékie-Slovaquie"
tabenspays(20, 2) = ""
tabenspays(20, 3) = ""
tabenspays(20, 4) = ""
tabenspays(20, 5) = ""
```

```
tabenspays(21, 1) = "Hongrie"
tabenspays(21, 2) = "Hongria"
tabenspays(21, 3) = ""
tabenspays(21, 4) = ""
tabenspays(21, 5) = ""
```

```
tabenspays(22, 1) = "Roumanie"
tabenspays(22, 2) = "Roumania"
tabenspays(22, 3) = ""
tabenspays(22, 4) = ""
tabenspays(22, 5) = ""
```

```
tabenspays(23, 1) = "Bulgarie"
tabenspays(23, 2) = "Bulgaria"
tabenspays(23, 3) = ""
tabenspays(23, 4) = ""
tabenspays(23, 5) = ""
```

```
tabenspays(24, 1) = "Serbie-Croatie"
tabenspays(24, 2) = "Yougoslavia"
tabenspays(24, 3) = ""
tabenspays(24, 4) = ""
tabenspays(24, 5) = ""
```

```
tabenspays(25, 1) = "Albanie"
tabenspays(25, 2) = "Albania"
tabenspays(25, 3) = ""
tabenspays(25, 4) = ""
tabenspays(25, 5) = ""
```

'Tableau des drapeaux pour le changement de langue

```
tabflag(1) = chem + "telemaqu\dessin\ffrance.bmp"
tabflag(2) = chem + "telemaqu\dessin\fbritain.bmp"
tabflag(3) = chem + "telemaqu\dessin\fgerman.bmp"
tabflag(4) = chem + "telemaqu\dessin\fitaly.bmp"
tabflag(5) = chem + "telemaqu\dessin\fspain.bmp"
```

'écran MenuPrinc

```
tabUnifPart(1) = "Universités partenaires"
tabUnifPart(2) = "Affiliated universities"
tabUnifPart(3) = "Universitäts-Partner"
tabUnifPart(4) = "Università consociate"
tabUnifPart(5) = "Universidades de intercambio"
```

'écran MenuUnifPart

```
tabchoixregion(1) = "Choisissez la région"
tabchoixregion(2) = "Choose the region"
tabchoixregion(3) = ""
tabchoixregion(4) = "Sceguete la regione"
tabchoixregion(5) = "Escoja la región"
```

```
tabchoixpays(1) = "Choisissez le pays"
tabchoixpays(2) = "Choose the country"
tabchoixpays(3) = ""
tabchoixpays(4) = "Sceguete il paese"
tabchoixpays(5) = "Escoja el país"
```

```
tabpc(1) = "Pays choisi :"
tabpc(2) = "Chosen country :"
tabpc(3) = ":"
tabpc(4) = "Paese scelto :"
tabpc(5) = "País escogido :"
```


'écran MenuVille

```
tabpresville(1) = "Présentation de la ville"
tabpresville(2) = "Presentation of the city"
tabpresville(3) = "Stadt Vorstellung"
tabpresville(4) = "Presentazione della città"
tabpresville(5) = "Presentación de la ciudad"
```

```
tabinfunif(1) = "Informations sur l'université"
tabinfunif(2) = "Information on the university"
tabinfunif(3) = "Informationen über die Universität"
tabinfunif(4) = "Informazioni sull'università"
tabinfunif(5) = "Información sobre la universidad"
```

```
tabinfrprat(1) = "Informations pratiques"
tabinfrprat(2) = "Practical information"
tabinfrprat(3) = "Praktische Informationen"
tabinfrprat(4) = "Informazioni pratiche"
tabinfrprat(5) = "Información práctica"
```

'Ecran MenuUnif

```
tabpresunif(1) = "Présentation de l'université"
tabpresunif(2) = "Presentation of the university"
tabpresunif(3) = "Universität Vorstellung"
tabpresunif(4) = "Presentazione dell'università"
tabpresunif(5) = "Presentación de la universidad"
```

```
tabchiffres(1) = "Quelques chiffres"
tabchiffres(2) = "Some figures"
tabchiffres(3) = "Ein paar Ziffern"
tabchiffres(4) = "Alcuni dati"
tabchiffres(5) = "Algunas cifras"
```

```
tabdepiecs(1) = "Département"
tabdepiecs(2) = "Faculty"
tabdepiecs(3) = "Abteilung"
tabdepiecs(4) = "Facoltà"
tabdepiecs(5) = "Facultades y escuelas universitarias"
```

'Ecran MenuIECS

```
tabcurs(1) = "Cursus"
tabcurs(2) = "Course structure"
tabcurs(3) = "Kursus"
tabcurs(4) = "Indirizzo"
tabcurs(5) = "Cursos"
```

```
taborga(1) = "Organisation"
taborga(2) = "Organisation"
taborga(3) = "Organisazion"
taborga(4) = "Organizzazione"
taborga(5) = "Organización"
```

```
tabcondad(1) = "Conditions d'admission"
tabcondad(2) = "Entry requirements"
tabcondad(3) = "Eintritts Bedingungen"
tabcondad(4) = "Condizioni di ammissione"
tabcondad(5) = "Condiciones de admisión"
```

'Ecran MenuInfo

```
tabadrut(1) = "Culture et Loisirs"
tabadrut(2) = "Culture and leisure"
tabadrut(3) = "Kultur und Freizeit"
tabadrut(4) = "Cultura e divertimenti"
tabadrut(5) = "Cultura y ocio"
```

```
tabviequot(1) = "Divers"
tabviequot(2) = "Others"
tabviequot(3) = "Andere"
tabviequot(4) = "Varie"
tabviequot(5) = "Varios"
```

```
tabplan(1) = "Plan de la ville"
```

DECLARE.BAS - 15

tabplan(2) = "Layout of the city"
tabplan(3) = "Plan von der Stadt"
tabplan(4) = "Pianta della città"
tabplan(5) = "Plano de la ciudad"

tabcalend(1) = "Dates à retenir"
tabcalend(2) = "Important dates"
tabcalend(3) = "Wichtige Datum"
tabcalend(4) = "Date da ricordare"
tabcalend(5) = "Fechas a recordar"

'Ecran MnuAut1

tabAUUnif(1) = "Bibliothèques"
tabAUUnif(2) = "Libraries"
tabAUUnif(3) = "Bibliothek"
tabAUUnif(4) = "Biblioteca"
tabAUUnif(5) = "Bibliotecas"

tabAUSecr(1) = "Musées"
tabAUSecr(2) = "Museums"
tabAUSecr(3) = "Museum"
tabAUSecr(4) = "Musei"
tabAUSecr(5) = "Museos"

tabAUCentre(1) = "Cinémas"
tabAUCentre(2) = "Cinemas"
tabAUCentre(3) = "Kinos"
tabAUCentre(4) = "Cinema"
tabAUCentre(5) = "Cines"

tabAUCentre2(1) = "Théâtres"
tabAUCentre2(2) = "Theatres"
tabAUCentre2(3) = "Theater"
tabAUCentre2(4) = "Teatri"
tabAUCentre2(5) = "Teatros"

tabAUBiblio(1) = "Sport"
tabAUBiblio(2) = "Sport"
tabAUBiblio(3) = "Sport"
tabAUBiblio(4) = "Sport"
tabAUBiblio(5) = "Deportes"

tabAUAssocEt(1) = "Associations d'étudiants"
tabAUAssocEt(2) = "Student societies"
tabAUAssocEt(3) = "Studenten Vereinigungen"
tabAUAssocEt(4) = "Associazioni studentesche"
tabAUAssocEt(5) = "Asociaciones estudiantiles"

'écrans MenuVieQuot

tabVQLog(1) = "Logements"
tabVQLog(2) = "Accommodation"
tabVQLog(3) = "Wohnungen"
tabVQLog(4) = "Alloggi"
tabVQLog(5) = "Alojamiento"

tabVQRestaus(1) = "Restaurants"
tabVQRestaus(2) = "Restaurants"
tabVQRestaus(3) = "Gaststätten"
tabVQRestaus(4) = "Ristoranti"
tabVQRestaus(5) = "Restaurantes"

tabVQBanques(1) = "Banques"
tabVQBanques(2) = "Banks"
tabVQBanques(3) = "Banken"
tabVQBanques(4) = "Banche"
tabVQBanques(5) = "Bancos"

tabVQPostes(1) = "Postes"
tabVQPostes(2) = "Postal services"
tabVQPostes(3) = "Postämte"
tabVQPostes(4) = "Uffici postali"
tabVQPostes(5) = "Oficinas de correos"

```

tabVQCinTh(1) = "A qui s'adresser"
tabVQCinTh(2) = "Licensed interlocutors"
tabVQCinTh(3) = "Redners privilegiert"
tabVQCinTh(4) = "A chi rivolgersi"
tabVQCinTh(5) = "Interlocutores privilegiados"

```

```

tabVQSport(1) = "Adresses"
tabVQSport(2) = "Addresses"
tabVQSport(3) = "Adresse"
tabVQSport(4) = "Indirizzi"
tabVQSport(5) = "Direcciones"

```

```
'son
```

```

tabhlp1(1) = chem + "telemaqu\son\français\hlp1.wav"
tabhlp1(2) = chem + "telemaqu\son\anglais\hlp1_ang.wav"
tabhlp1(3) = chem + "telemaqu\son\allemand\hlp1_all.wav"
tabhlp1(4) = chem + "telemaqu\son\italien\hlp1_ita.wav"
tabhlp1(5) = chem + "telemaqu\son\espagnol\hlp1_esp.wav"

```

```

tabhlp2(1) = chem + "telemaqu\son\français\hlp2.wav"
tabhlp2(2) = chem + "telemaqu\son\anglais\hlp2_ang.wav"
tabhlp2(3) = chem + "telemaqu\son\allemand\hlp2_all.wav"
tabhlp2(4) = chem + "telemaqu\son\italien\hlp2_ita.wav"
tabhlp2(5) = chem + "telemaqu\son\espagnol\hlp2_esp.wav"

```

```

tabhist(1) = chem + "telemaqu\son\français\hist.wav"
tabhist(2) = chem + "telemaqu\son\anglais\hist_ang.wav"
tabhist(3) = chem + "telemaqu\son\allemand\hist_all.wav"
tabhist(4) = chem + "telemaqu\son\italien\hist_ita.wav"
tabhist(5) = chem + "telemaqu\son\espagnol\hist_esp.wav"

```

```

tablang(1) = chem + "telemaqu\son\français\lang.wav"
tablang(2) = chem + "telemaqu\son\anglais\lang_ang.wav"
tablang(3) = chem + "telemaqu\son\allemand\lang_all.wav"
tablang(4) = chem + "telemaqu\son\italien\lang_ita.wav"
tablang(5) = chem + "telemaqu\son\espagnol\lang_esp.wav"

```

```

tabmenu(1) = chem + "telemaqu\son\français\menu.wav"
tabmenu(2) = chem + "telemaqu\son\anglais\menu_ang.wav"
tabmenu(3) = chem + "telemaqu\son\allemand\menu_all.wav"
tabmenu(4) = chem + "telemaqu\son\italien\menu_ita.wav"
tabmenu(5) = chem + "telemaqu\son\espagnol\menu_esp.wav"

```

```

tabprnt(1) = chem + "telemaqu\son\français\prnt.wav"
tabprnt(2) = chem + "telemaqu\son\anglais\prnt_ang.wav"
tabprnt(3) = chem + "telemaqu\son\allemand\prnt_all.wav"
tabprnt(4) = chem + "telemaqu\son\italien\prnt_ita.wav"
tabprnt(5) = chem + "telemaqu\son\espagnol\prnt_esp.wav"

```

```

tabquit(1) = chem + "telemaqu\son\français\quit.wav"
tabquit(2) = chem + "telemaqu\son\anglais\quit_ang.wav"
tabquit(3) = chem + "telemaqu\son\allemand\quit_all.wav"
tabquit(4) = chem + "telemaqu\son\italien\quit_ita.wav"
tabquit(5) = chem + "telemaqu\son\espagnol\quit_esp.wav"

```

```

tabreto(1) = chem + "telemaqu\son\français\reto.wav"
tabreto(2) = chem + "telemaqu\son\anglais\reto_ang.wav"
tabreto(3) = chem + "telemaqu\son\allemand\reto_all.wav"
tabreto(4) = chem + "telemaqu\son\italien\reto_ita.wav"
tabreto(5) = chem + "telemaqu\son\espagnol\reto_esp.wav"

```

```

tabunif(1) = chem + "telemaqu\son\français\unif.wav"
tabunif(2) = chem + "telemaqu\son\anglais\unif_ang.wav"
tabunif(3) = chem + "telemaqu\son\allemand\unif_all.wav"
tabunif(4) = chem + "telemaqu\son\italien\unif_ita.wav"
tabunif(5) = chem + "telemaqu\son\espagnol\unif_esp.wav"

```

```

tabpl(1) = chem + "telemaqu\son\français\plan.wav"
tabpl(2) = chem + "telemaqu\son\anglais\plan_ang.wav"

```


DECLARE.BAS - 17

```
tabpl(3) = chem + "telemaqu\son\allemand\plan_all.wav"  
tabpl(4) = chem + "telemaqu\son\italien\plan_ita.wav"  
tabpl(5) = chem + "telemaqu\son\espagnol\plan_esp.wav"
```

```
tabstat(1) = chem + "telemaqu\graph\stat.xls!graph1"  
tabstat(2) = chem + "telemaqu\graph\stat1.xls!graph1"
```

'Tableau pour les labels

```
tablbl(1, 2) = "Menu Principal"  
tablbl(2, 2) = "Main Menu"  
tablbl(3, 2) = "Haupt Menu"  
tablbl(4, 2) = "Menu principale "  
tablbl(5, 2) = "Menú Principal"
```

```
Dim cptab As Integer  
cptab = 1
```

While cptab <= nbl

```
tablbl(cptab, 4) = tabinfunif(cptab)  
tablbl(cptab, 5) = tabdepieces(cptab)  
tablbl(cptab, 6) = tabinfprat(cptab)  
tablbl(cptab, 7) = tabviequot(cptab)  
tablbl(cptab, 8) = tabadrut(cptab)  
tablbl(cptab, 10) = tabcalend(cptab)  
tablbl(cptab, 11) = tabcondad(cptab)  
tablbl(cptab, 12) = tabcurs(cptab)  
tablbl(cptab, 14) = taborga(cptab)  
tablbl(cptab, 15) = tabplan(cptab)  
tablbl(cptab, 18) = tabchiffres(cptab)  
tablbl(cptab, 19) = tabUnifPart(cptab)  
cptab = cptab + 1  
Wend
```

'Message de non-disponibilité

```
tabmess(1) = "Pas d'information disponible pour l'instant"  
tabmess(2) = "No information available at this moment"  
tabmess(3) = ""  
tabmess(4) = ""  
tabmess(5) = ""
```

```
tabtitremess(1) = "Information"  
tabtitremess(2) = "Information"  
tabtitremess(3) = "Information"  
tabtitremess(4) = "Informazione"  
tabtitremess(5) = "Información"
```

'Message de non-permission d'impression

```
tabimpression(1) = "Vous n'avez plus le droit d'imprimer"  
tabimpression(2) = "You may not print anymore"  
tabimpression(3) = ""  
tabimpression(4) = ""  
tabimpression(5) = ""
```

```
tabtitreimpr(1) = "Attention"  
tabtitreimpr(2) = "Attention"  
tabtitreimpr(3) = "Achtung"  
tabtitreimpr(4) = ""  
tabtitreimpr(5) = ""
```

'Remplissage du dynaset ds contenant tous les pays présents
'dans la Base de données (redondant mais utile si on vient
'de l'écran suivant)

```
MDI1!Data1.RecordSource = "select NOM_P from PAYS"  
MDI1!Data1.Refresh  
Set ds = MDI1!Data1.Recordset
```

k% = 1

While Not (ds.EOF)

 i% = 1

 tr% = False

 While (i% <= UBound(tabenspays)) And (tr% = False)

 If tabenspays(i%, 1) = ds("NOM_P") Then

DECLARE.BAS - 18

```
        tr% = True
        ReDim Preserve tabpaysdans(k%)
        tabpaysdans(k%) = i%
        k% = k% + 1
    Else
        i% = i% + 1
    End If
Wend
ds.MoveNext
Wend

'tableau de fichiers MIDI
repert = Dir(chem + "telemaqu\son\midi\*.mid")
longtabmidi = 0
While (repert <> "") And (longtabmidi <= UBound(tabmidi))
    longtabmidi = longtabmidi + 1
    tabmidi(longtabmidi) = repert
    repert = Dir
Wend

'tableau de fichiers WAVE
repert = Dir(chem + "telemaqu\son\wave\*.wav")
longtabwave = 0
While (repert <> "") And (longtabwave <= UBound(tabwave))
    longtabwave = longtabwave + 1
    tabwave(longtabwave) = repert
    repert = Dir
Wend

'Initialisation du générateur de nombres aléatoires
Randomize

End Sub

Sub initintroville ()
If chgtl = False Then

    If ds_Ville("PHRASEFRA_V") <> "#NULL#" Then
        tabintroville(1) = ds_Ville("PHRASEFRA_V")
        introville.lblsstitre.Visible = True
    End If
    If ds_Ville("PHRASEANG_V") <> "#NULL#" Then
        tabintroville(2) = ds_Ville("PHRASEANG_V")
        introville.lblsstitre.Visible = True
    End If
    If ds_Ville("PHRASEALL_V") <> "#NULL#" Then
        tabintroville(3) = ds_Ville("PHRASEALL_V")
        introville.lblsstitre.Visible = True
    End If
    If ds_Ville("PHRASEESP_V") <> "#NULL#" Then
        tabintroville(5) = ds_Ville("PHRASEESP_V")
        introville.lblsstitre.Visible = True
    End If
    If ds_Ville("PHRASEITA_V") <> "#NULL#" Then
        tabintroville(4) = ds_Ville("PHRASEITA_V")
        introville.lblsstitre.Visible = True
    End If
Else
    chgtl = False
End If
introville.lblsstitre.Caption = tabintroville(1)
End Sub

Sub initmenuiecs ()
MenuIECS.cmdcurs.Caption = tabcurs(1)
MenuIECS.cmdorga.Caption = taborga(1)
MenuIECS.cmdcondad.Caption = tabcondad(1)
End Sub

Sub InitMenuInfos ()
MenuInfos.CmdAdrUt.Caption = tabadrut(1)
MenuInfos.CmdVieQuot.Caption = tabviequot(1)
MenuInfos.CmdPlan.Caption = tabplan(1)
```

DECLARE.BAS - 19

```
MenuInfos.Cmdcalend.Caption = tabcalend(1)
MenuInfos.lblsstitre.Caption = tablbl(1, ecrancour)
End Sub
```

```
Sub initmenupays ()
  MenuPays.lbltitre = tabenspays(pays_ok, 1)
  chgtl = False
End Sub
```

```
Sub initmenuprinc ()

menuprinc.cmdUnifPart.Caption = tabUnifPart(1)
menuprinc.lblsstitre.Caption = tablbl(1, ecrancour)
End Sub
```

```
Sub initmenuunif ()

'initialisation des boutons
MenuUnif.cmdpresunif.Caption = tabpresunif(1)
MenuUnif.cmdchiffres.Caption = tabchiffres(1)
MenuUnif.cmddepiecs.Caption = tabdepiecs(1)

'Initialisation du sous-titre
MenuUnif.lblsstitre.Caption = tablbl(1, ecrancour)

End Sub
```

```
Sub initMenuUnifPart ()
MenuUnifPart.lbltitre.Caption = tablbl(1, ecrancour)
MenuUnifPart.lblchoixregion.Caption = tabchoixregion(1)
MenuUnifPart.lblchoixpays.Caption = tabchoixpays(1)
MenuUnifPart.lblpc.Caption = tabpc(1)
MenuUnifPart.lblpays.Caption = tabenspays(pays_ok, 1)
chgtl = False

End Sub
```

```
Sub InitMenuVieQuot ()
MenuVieQuot.CmdVQLog.Caption = tabVQLog(1)
MenuVieQuot.CmdVQRestaus.Caption = tabVQRestaus(1)
MenuVieQuot.CmdVQBanques.Caption = tabVQBanques(1)
MenuVieQuot.CmdVQPostes.Caption = tabVQPostes(1)
MenuVieQuot.CmdVQCinTh.Caption = tabVQCinTh(1)
MenuVieQuot.CmdVQSport.Caption = tabVQSport(1)
MenuVieQuot.lblsstitre.Caption = tablbl(1, ecrancour)
End Sub
```

```
Sub initmenuville ()
menuville.cmdPresVille.Caption = tabpresville(1)
menuville.cmdInfUnif.Caption = tabinfunif(1)
menuville.cmdInfPrat.Caption = tabinfprat(1)
chgtl = False
End Sub
```

```
Sub InitMnuAut1 ()
MnuAut1.CmdAUUnif.Caption = tabAUUnif(1)
MnuAut1.cmdAUSecr.Caption = tabAUSecr(1)
MnuAut1.CmdAUCentre.Caption = tabAUCentre(1)
MnuAut1.CmdCinTh2.Caption = tabAUCentre2(1)
MnuAut1.CmdAUBiblio.Caption = tabAUBiblio(1)
MnuAut1.CmdAUAssocEt.Caption = tabAUAssocEt(1)
MnuAut1.lblsstitre.Caption = tablbl(1, ecrancour)
End Sub
```

```
Sub InitMnuAut2 ()

End Sub
```

```
Sub InitOrga ()
Dim sql As String
Dim ds_orga As dynaset
Const nbre = 4
orga.lblsstitre.Caption = taborga(1)
'Accès bd pour afficher le logo universitaire
MDI1!Data1.RecordSource = "select LOGO_U from UNIVERSITE WHERE NOM_P = '" + req_pays + "' and NO
```


DECLARE.BAS - 20

```
V="" + req_ville + "" and NOM_U="" + req_univ + ""
MDI1!Data1.Refresh
Set ds_unif = MDI1!Data1.Recordset
If ds_unif("LOGO_U") <> "#NULL#" Then
    On Error GoTo errorhandlerorga
    orga!Picture1.Picture = LoadPicture(chem + "TELEMAQU\" + Left$(req_pays, 8) + "\" + Left$(req_
ille, 8) + "\UNIV\" + ds_unif("LOGO_U") + ".bmp")
    orga!Picture1.Visible = True
End If
un_orga:
'Accès pour obtenir les infos pour l'organisation
sql = " Select FONCFRA_O, FONCANG_O, FONCALL_O, FONCITA_O, FONCESP_O, NOM_O "
sql = sql + " from ORGANISATION WHERE "
sql = sql + " NOM_P = '" + req_pays + "' and NOM_V='" + req_ville + "' and NOM_U='"
sql = sql + req_univ + "' and NOM_F='" + req_fac + "' order by FONCFRA_O"
MDI1!Data1.RecordSource = sql
MDI1!Data1.Refresh
Set ds_orga = MDI1!Data1.Recordset
If ds_orga.EOF = False Then
    ds_orga.MoveLast
    nbrorga = ds_orga.RecordCount
    ReDim taborgafct(ds_orga.RecordCount, nbl)
    ReDim taborgan(ds_orga.RecordCount)
    i% = 1
    ds_orga.MoveFirst
    While Not (ds_orga.EOF)
        If ds_orga("FONCFRA_O") <> "#NULL#" Then taborgafct(i%, 1) = ds_orga("FONCFRA_O")
        If ds_orga("FONCANG_O") <> "#NULL#" Then taborgafct(i%, 2) = ds_orga("FONCANG_O")
        If ds_orga("FONCALL_O") <> "#NULL#" Then taborgafct(i%, 3) = ds_orga("FONCALL_O")
        If ds_orga("FONCITA_O") <> "#NULL#" Then taborgafct(i%, 4) = ds_orga("FONCITA_O")
        If ds_orga("FONCESP_O") <> "#NULL#" Then taborgafct(i%, 5) = ds_orga("FONCESP_O")
        If ds_orga("NOM_O") <> "#NULL#" Then taborgan(i%) = ds_orga("NOM_O")
        i% = i% + 1
    ds_orga.MoveNext
    Wend
    cporga = 1
    compteur = 0

    While (cporga <= nbrorga) And (compteur < nbre)
        orga!lblfonc(compteur).Caption = taborgafct(cporga, 1)
        orga!lblnom(compteur).Caption = taborgan(cporga)
        orga!lblfonc(compteur).Visible = True
        orga!lblnom(compteur).Visible = True
        cporga = cporga + 1
        compteur = compteur + 1
    Wend

    If cporga <= nbrorga Then
        orga!suiv.Visible = True
    End If
    cporga = cporga - compteur
Else
    afficher
    tabidform(tabprecform(ecrancour)).Show
End If
Exit Sub
errorhandlerorga:
Resume un_orga
End Sub

Sub InitOrgabis ()
Const nbre = 4
compteur = 0
orga!lblsstitre.Caption = taborga(1)
While (cporga <= nbrorga) And (compteur < nbre)
    orga!lblfonc(compteur).Caption = taborgafct(cporga, 1)
    orga!lblnom(compteur).Caption = taborgan(cporga)
    orga!lblfonc(compteur).Visible = True
    orga!lblnom(compteur).Visible = True
    cporga = cporga + 1
    compteur = compteur + 1
Wend

If cporga <= nbrinfos Then
    orga!suiv.Visible = True
```

DECLARE.BAS - 21

```
End If
cporga = cporga - compteur
```

End Sub

```
Sub InitPlans ()
plans.lblsstitre = tablbl(1, ecrancour)
End Sub
```

```
Sub initpresunif1 ()
```

End Sub

```
Sub initstat1 ()
Dim sql As String
Dim compt As Integer
Dim a As String
```

```
Stat1.lblsstitre.Caption = tablbl(1, ecrancour)
'recherche de tous les graphes concernant une université
sql = "select NOMFRA_G, NOMANG_G, NOMALL_G, NOMITA_G, NOMESP_G, TYPE_G, ABSCISSE_G"
sql = sql + " from GRAPHIQUE WHERE NOM_P = '" + req_pays + "'" and NOM_V = '" + req_ville
sql = sql + "'" and NOM_U = '" + req_univ + "'" ORDER BY NOMFRA_G"
```

```
MDI1!Data1.RecordSource = sql
```

```
MDI1!Data1.Refresh
```

```
Set ds_graf = MDI1!Data1.Recordset
```

```
If ds_graf.EOF = False Then
```

```
ds_graf.MoveLast
```

```
ReDim tabg(ds_graf.RecordCount)
```

```
ds_graf.MoveFirst
```

```
While ds_graf.EOF = False
```

```
If ds_graf("NOMFRA_G") <> "#NULL#" Then tabg(comptgraf).tabnomgraf(1) = ds_graf("NOMFRA_G")
```

```
If ds_graf("NOMANG_G") <> "#NULL#" Then tabg(comptgraf).tabnomgraf(2) = ds_graf("NOMANG_G")
```

```
If ds_graf("NOMALL_G") <> "#NULL#" Then tabg(comptgraf).tabnomgraf(3) = ds_graf("NOMALL_G")
```

```
If ds_graf("NOMITA_G") <> "#NULL#" Then tabg(comptgraf).tabnomgraf(4) = ds_graf("NOMITA_G")
```

```
If ds_graf("NOMESP_G") <> "#NULL#" Then tabg(comptgraf).tabnomgraf(5) = ds_graf("NOMESP_G")
```

```
If ds_graf("ABSCISSE_G") <> "#NULL#" Then
tabg(comptgraf).typ = ds_graf("TYPE_G")
```

```
Else
```

```
tabg(comptgraf).typ = "Histogramme"
```

```
End If
```

```
tabg(comptgraf).absc = ds_graf("ABSCISSE_G")
```

```
ds_graf.MoveNext
```

```
comptgraf = comptgraf + 1
```

```
Wend
```

```
'On démarre avec le premier graphe
```

```
comptgraf = 1
```

```
initstatlbis
```

```
Else
```

```
ReDim tabg(1)
```

```
afficher
```

```
MenuUnif.Show
```

```
End If
```

End Sub

```
Sub initstatlbis ()
```

```
Dim compt, a, o, h As Integer
```

```
Dim prov As String
```

```
Dim tababsc() As Integer
```

```
Dim tabordo() As tordo
```

```
'Requête pour trouver tous les champs d'un graphe
```

```
sql = "select CHAMPFRA_G, CHAMPANG_G, CHAMPALL_G, CHAMPITA_G, CHAMPESP_G, ORDONNEE_G"
```

```
sql = sql + " from CHIFFRES WHERE NOM_P = '" + req_pays + "'" and NOM_V = '" + req_ville
```

```
sql = sql + "'" and NOM_U = '" + req_univ + "'" and NOMFRA_G = '" + tabg(comptgraf).tabnomgraf(1)
```

```
sql = sql + "'" ORDER BY CHAMPFRA_G"
```

```
MDI1!Data1.RecordSource = sql
```

```
MDI1!Data1.Refresh
```

```
Set ds_chiffres = MDI1!Data1.Recordset
```

```
If ds_chiffres.EOF = False Then
```

```
Stat1.Graphel.Visible = True
```

DECLARE.BAS - 22

```
ds_chiffres.MoveLast
ReDim tabchiffre(ds_chiffres.RecordCount)
ds_chiffres.MoveFirst
compt = 1
While ds_chiffres.EOF = False
    If ds_chiffres("CHAMPFRA_G") <> "#NULL#" Then tabchiffre(compt).nomch(1) = ds_chiffres("CHAMPFRA_G")
    If ds_chiffres("CHAMPANG_G") <> "#NULL#" Then tabchiffre(compt).nomch(2) = ds_chiffres("CHAMPANG_G")
    If ds_chiffres("CHAMPALL_G") <> "#NULL#" Then tabchiffre(compt).nomch(3) = ds_chiffres("CHAMPALL_G")
    If ds_chiffres("CHAMPITA_G") <> "#NULL#" Then tabchiffre(compt).nomch(4) = ds_chiffres("CHAMPITA_G")
    If ds_chiffres("CHAMPESP_G") <> "#NULL#" Then tabchiffre(compt).nomch(5) = ds_chiffres("CHAMPESP_G")
    tabchiffre(compt).ordo = ds_chiffres("ORDONNEE_G")
    ds_chiffres.MoveNext
    compt = compt + 1
Wend
'Transformation des strings abscisse et ordonnée en 2 tableaux d'entiers

'Abscisse
k = 1
compt = 1
While k <= Len(tabg(comptgraf).absc)
    If Mid(tabg(comptgraf).absc, k, 1) = "," Then
        compt = compt + 1
    End If
    k = k + 1
Wend
ReDim tababsc(compt)
k = 1
compt = 1
prov = ""
While k <= Len(tabg(comptgraf).absc)
    If Mid(tabg(comptgraf).absc, k, 1) = "," Then
        tababsc(compt) = Val(prov)
        prov = ""
        compt = compt + 1
    Else
        prov = prov + Mid(tabg(comptgraf).absc, k, 1)
    End If
    k = k + 1
Wend
tababsc(compt) = Val(prov)

'Ordonnée
ReDim tabordo(UBound(tabchiffre))
compt = 1
While compt <= UBound(tabchiffre)
    k = 1
    h = 1
    prov = ""
    While k <= Len(tabchiffre(compt).ordo)
        If Mid(tabchiffre(compt).ordo, k, 1) = "," Then
            tabordo(compt).valeurs(h) = Val(prov)
            prov = ""
            h = h + 1
        Else
            prov = prov + Mid(tabchiffre(compt).ordo, k, 1)
        End If
        k = k + 1
    End While
    tabordo(compt).valeurs(h) = Val(prov)
    compt = compt + 1
Wend

'Dessin du graphe

'reset
'stat1.Graphel.DataReset = 1

'Style du graphe
```


DECLARE.BAS - 23

```
Select Case tabg(comptgraf).typ
    Case "Histogramme"
        Stat1.Graphe1.GraphType = 4
    Case "Lignes"
        Stat1.Graphe1.GraphType = 6
    Case "Aires"
        Stat1.Graphe1.GraphType = 8
End Select

'Nombre de champs
Stat1.Graphe1.NumSets = UBound(tabchiffre)
'Nombre d'abscisses
Stat1.Graphe1.NumPoints = UBound(tababsc)
'Placement des valeurs
For a = 1 To Stat1.Graphe1.NumSets
    Stat1.Graphe1.ThisSet = a
    For o = 1 To Stat1.Graphe1.NumPoints
        Stat1.Graphe1.ThisPoint = o
        Stat1.Graphe1.GraphData = tabordo(a).valeurs(o)
    Next o
Next a

'Texte des abscisses
For o = 1 To Stat1.Graphe1.NumPoints
    Stat1.Graphe1.ThisPoint = o
    Stat1.Graphe1.LabelText = tababsc(o)
Next o

initstat1cl

Else
Stat1.Graphe1.Visible = False
afficher
MenuUnif.Show
End If

End Sub

Sub initstat1cl ()
Dim a As Integer

'Initialisation du sous-titre
Stat1.lblsstitre.Caption = tablbl(1, ecrancour)

'titre du graphe (en bas)
Stat1.Graphe1.BottomTitle = tabg(comptgraf).tabnomgraf(1)

'Légende des champs
For a = 1 To Stat1.Graphe1.NumSets
    Stat1.Graphe1.ThisPoint = a
    Stat1.Graphe1.LegendText = tabchiffre(a).nomch(1)
Next a

'Mode du dessin
Stat1.Graphe1.DrawMode = 3

End Sub

Sub ListForm ()

Set tabidform(1) = Pres
tabprecform(1) = 0
Set tabidform(2) = menuprinc
tabprecform(2) = 1
Set tabidform(3) = menuville
tabprecform(3) = 2
Set tabidform(4) = MenuUnif
tabprecform(4) = 3
Set tabidform(5) = MenuIECS
tabprecform(5) = 4
Set tabidform(6) = MenuInfos
tabprecform(6) = 3
Set tabidform(7) = MenuVieQuot
tabprecform(7) = 6
```

```

Set tabidform(8) = MnuAUt1
tabprecform(8) = 6
Set tabidform(9) = MnuAUt2
tabprecform(9) = 6
Set tabidform(10) = Calend
tabprecform(10) = 6
Set tabidform(11) = CondAdm
tabprecform(11) = 5
Set tabidform(12) = cursus
tabprecform(12) = 5
Set tabidform(13) = cours
tabprecform(13) = 12
Set tabidform(14) = orga
tabprecform(14) = 5
Set tabidform(15) = plans
tabprecform(15) = 6
Set tabidform(16) = presunif1
tabprecform(16) = 4
Set tabidform(17) = presville
tabprecform(17) = 3
Set tabidform(18) = Stat1
tabprecform(18) = 4
Set tabidform(19) = MenuUnifPart
tabprecform(19) = 2
Set tabidform(20) = MenuPays
tabprecform(20) = 19
Set tabidform(21) = introville
tabprecform(21) = 0
Set tabidform(22) = MenuPays
tabprecform(22) = 19
Set tabidform(23) = AUBiblio
tabprecform(23) = 0
Set tabidform(24) = fhisto
tabprecform(24) = 0
Set tabidform(25) = ChgtUniv
tabprecform(25) = 0
Set tabidform(26) = IntroVillebis
tabprecform(26) = 20
End Sub

```

TRAITEME.BAS - 1

'Nombre d'impressions permises pour l'ensemble du logiciel
Global Const cimprt0t = 10

'Nombre d'impressions permises pour une matière
Global Const cimprloc = 3

'Nombre d'items imprimés par page
Global Const nbritems = 10

'Variable calculant le nombre d'impressions totales
'jusqu'à présent
Global nbrimprt0t As Integer

'Variable calculant le nombre d'impressions locales
'jusqu'à présent
Global nbrimprloc As Integer

Sub afficher ()
MsgBox tabmess(1), 64, tabtitremess(1)
End Sub

Sub affichimpr ()
MsgBox tabimpression(1), 48, tabtitreimpr(1)
End Sub

Sub guignols ()
Dim cpwave As Integer

If boolfin = False Then
 MDI1.MM1.Command = "stop"
 MDI1.MM1.Command = "close"
End If
DoEvents
boolfin = False
MDI1.MM1.Notify = False
MDI1.MM1.Wait = True
cpwave = Int(longtabwave * Rnd + 1)
MDI1.MM1.FileName = chem + "telemaqu\son\wave\" + tabwave(cpwave)
MDI1.MM1.Command = "open"
MDI1.MM1.Command = "play"
End Sub

Function improk () As Integer

If (nbrimprloc >= cimprloc) Or (nbrimprt0t >= cimprt0t) Then
 improk = False
 affichimpr
Else
 nbrimprloc = nbrimprloc + 1
 nbrimprt0t = nbrimprt0t + 1
 improk = True
End If
End Function

Sub prochisto ()

If prechisto = False Then
'pour capturer l'écran
 clipboard.Clear
 SendKeys "^{F1}"
 DoEvents
 If cpthist = 10 Then
 For i% = 2 To 9
 tabhisto(i% - 1).numero = tabhisto(i%).numero
 tabhisto(i% - 1).country = tabhisto(i%).country
 tabhisto(i% - 1).city = tabhisto(i%).city
 tabhisto(i% - 1).univer = tabhisto(i%).univer
 tabhisto(i% - 1).depart = tabhisto(i%).depart
 tabhisto(i% - 1).Langue = tabhisto(i%).Langue
 FHisto.cmdhisto(i% - 1).Picture = FHisto.cmdhisto(i%).Picture
 Next i%
 tabhisto(9).numero = ecranour
 tabhisto(9).country = req_pays
 tabhisto(9).city = req_ville

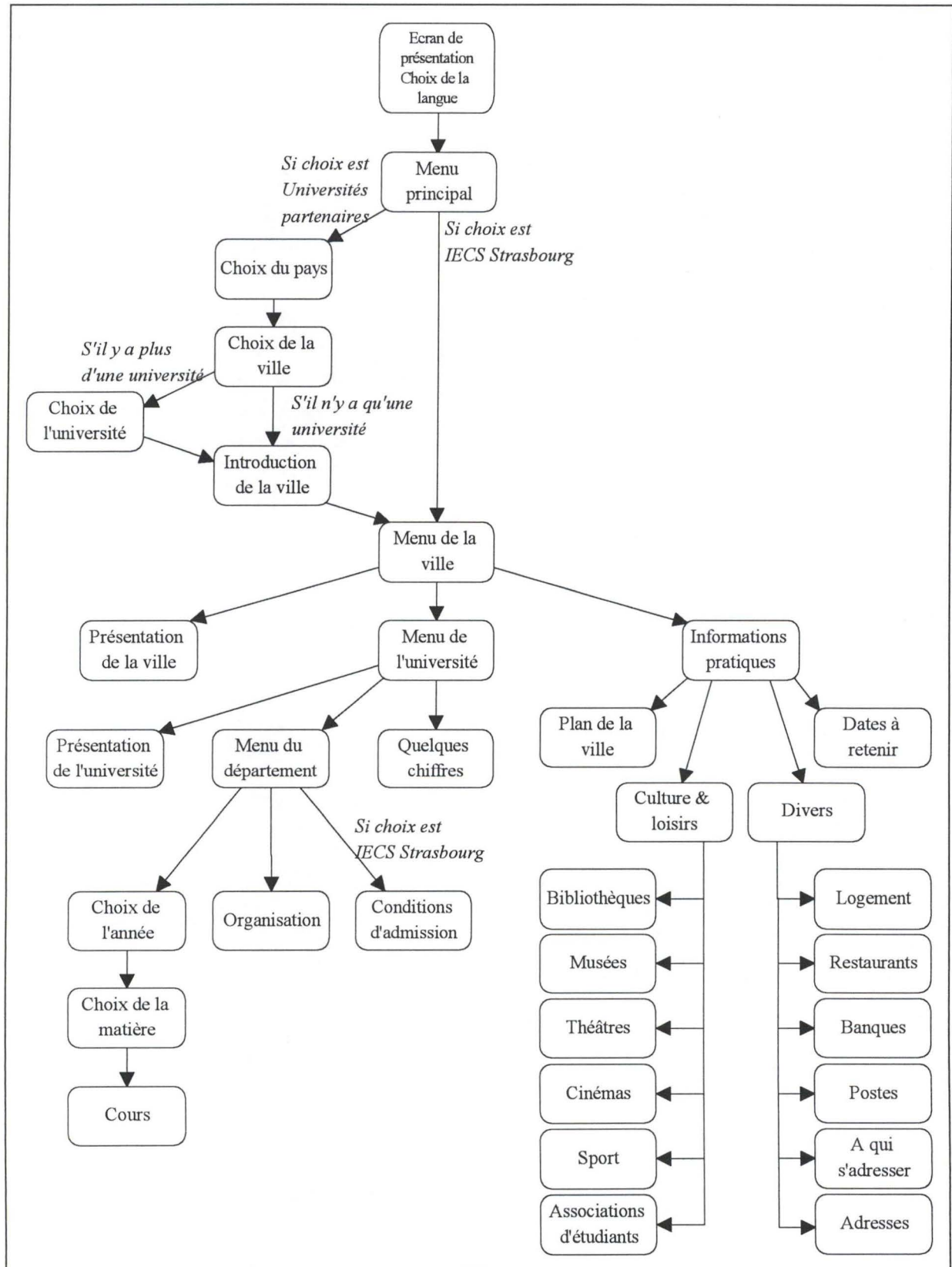

```

        tabhisto(9).univer = req_univ
        tabhisto(9).depart = req_fac
        tabhisto(9).Langue = 1
        FHisto.cmdhisto(9).Picture = clipboard.GetData()
    Else
        'Gestion de l'icône "historique" dans la barre d'outils
        temp$ = chem + "telemaqu\dessin\grille" + Right(Str(cpthist), 1) + ".bmp"
        MDI1.Histo.Picture = LoadPicture(temp$)
        tabhisto(cpthist).numero = ecrancour
        tabhisto(cpthist).country = req_pays
        tabhisto(cpthist).city = req_ville
        tabhisto(cpthist).univer = req_univ
        tabhisto(cpthist).depart = req_fac
        tabhisto(cpthist).Langue = 1
        FHisto.cmdhisto(cpthist).Picture = clipboard.GetData()
        cpthist = cpthist + 1
    End If
Else
    prechisto = False
End If
End Sub

```

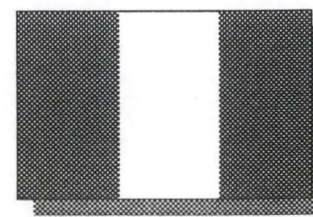
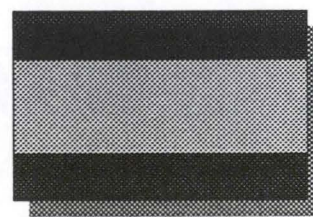
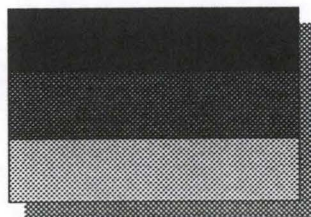
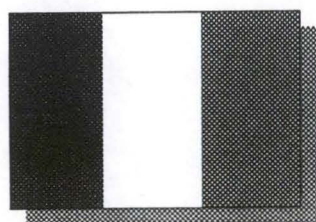
ANNEXE 2. SCÉNARIO DE TÉLÉMAQUE

Pour permettre un parcours plus aisé dans le scénario de Télémaque, nous recopions ici le graphe de transition. Celui-ci présente les liens fixes entre les différents écrans. Nous proposons, dans les pages qui suivent, la première partie du scénario. Nous prenons comme hypothèse que le choix de l'utilisateur au niveau du menu principal est: IECS Strasbourg.





TELEMAQUA





Menu Principal

IECS Strasbourg

Universités partenaires

Menu graphique



Strasbourg

Présentation de la ville

Informations sur l'université

Informations pratiques

Menu graphique



Strasbourg

Présentation de la ville

**ENTRE 2 ET 6 ECRANS POUR
L'HISTORIQUE DE LA VILLE**



Strasbourg

Présentation de la ville

**Entre 3 et 5 écrans pour définir
les curiosités culturelles**



Strasbourg

Présentation de la ville

**Au maximum 1 écran pour
présenter le rôle politique de la
ville**



Strasbourg

Présentation de la ville

Informations sur l'université

Informations pratiques

Menu graphique



IECS Strasbourg

Informations sur l'université



Présentation de l'université

Quelques chiffres

Département IECS

Menu graphique



JECS Strasbourg

Présentation de l'université



- Maximum 5 écrans pour présenter un historique de l'université (ex : ligne du temps) et sa situation actuelle (composition)
- Utilisation de texte, dessins, photos, vidéos, ...



IECS Strasbourg

Quelques chiffres



Statistiques sur

- ☎ l'évolution du nombre d'étudiants dans l'université
- ☎ l'évolution du nombre d'étudiants à l'IECS
- ☎ l'évolution du nombre d'étudiants dans chaque année à l'IECS
- ☎ le nombre d'étudiants par professeur
- ☎ l'évolution du nombre de diplômés

Menu graphique



Cursus

Organisation

Conditions d'admission

Menu graphique

Menu graphique

Quatrième année

Troisième année

Deuxième année

Première année

Cursus

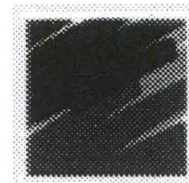
UCL5 Strasbourg





JECS Strasbourg

Première année



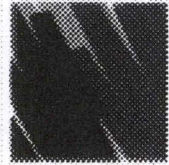
Comptabilité

Informatique

Droit

Gestion

Menu graphique



Informatique et société

J. Pallon

45 h

Menu graphique



JECS Strasbourg

Organisation



- Organigramme de la faculté
- Nom, fonction, photo des personnes y figurant
- présentation des professeurs + photos

Menu graphique

Menu graphique

Informations pratiques

Informations sur l'université

Présentation de la ville



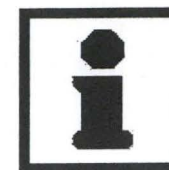
UNIVERSITÉ DE STRASBOURG





Strasbourg

Informations pratiques



Culture et loisirs

Divers

Plan de la ville

Dates à retenir

Menu graphique

Strasbourg

Culture et loisirs



Cinémas

Théâtres

Musées

Associations
d'étudiants

Sport

Bibliothèques

Menu graphique



Strasbourg

Divers



Logements

Restaurants

Banques

Postes

A qui s'adresser

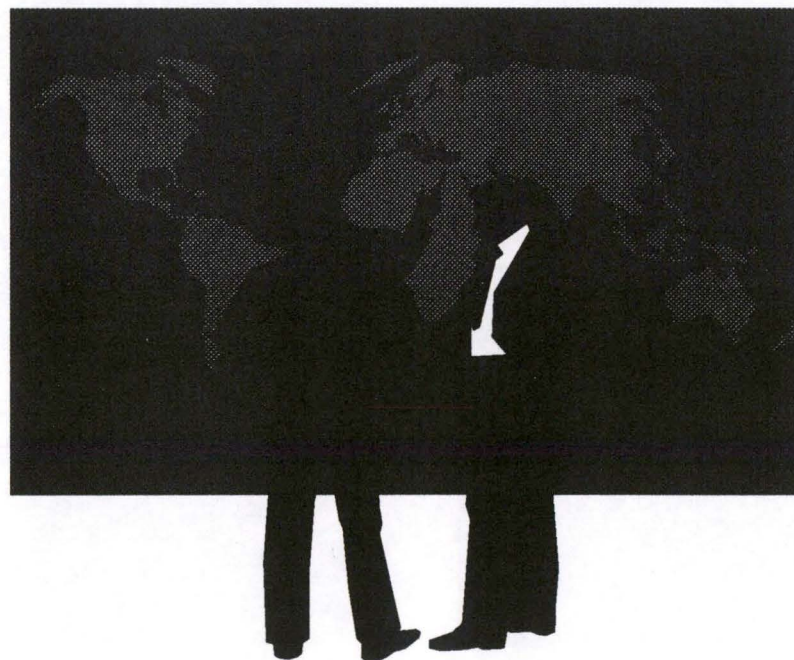
Adresses

Menu graphique



JECS Strasbourg

Plan de la ville



Menu graphique



JECs Strasbourg

Dates à retenir



Marché de Noël

Novembre et décembre

Place de la cathédrale

Menu graphique



J&CS Strasbourg

Restaurants

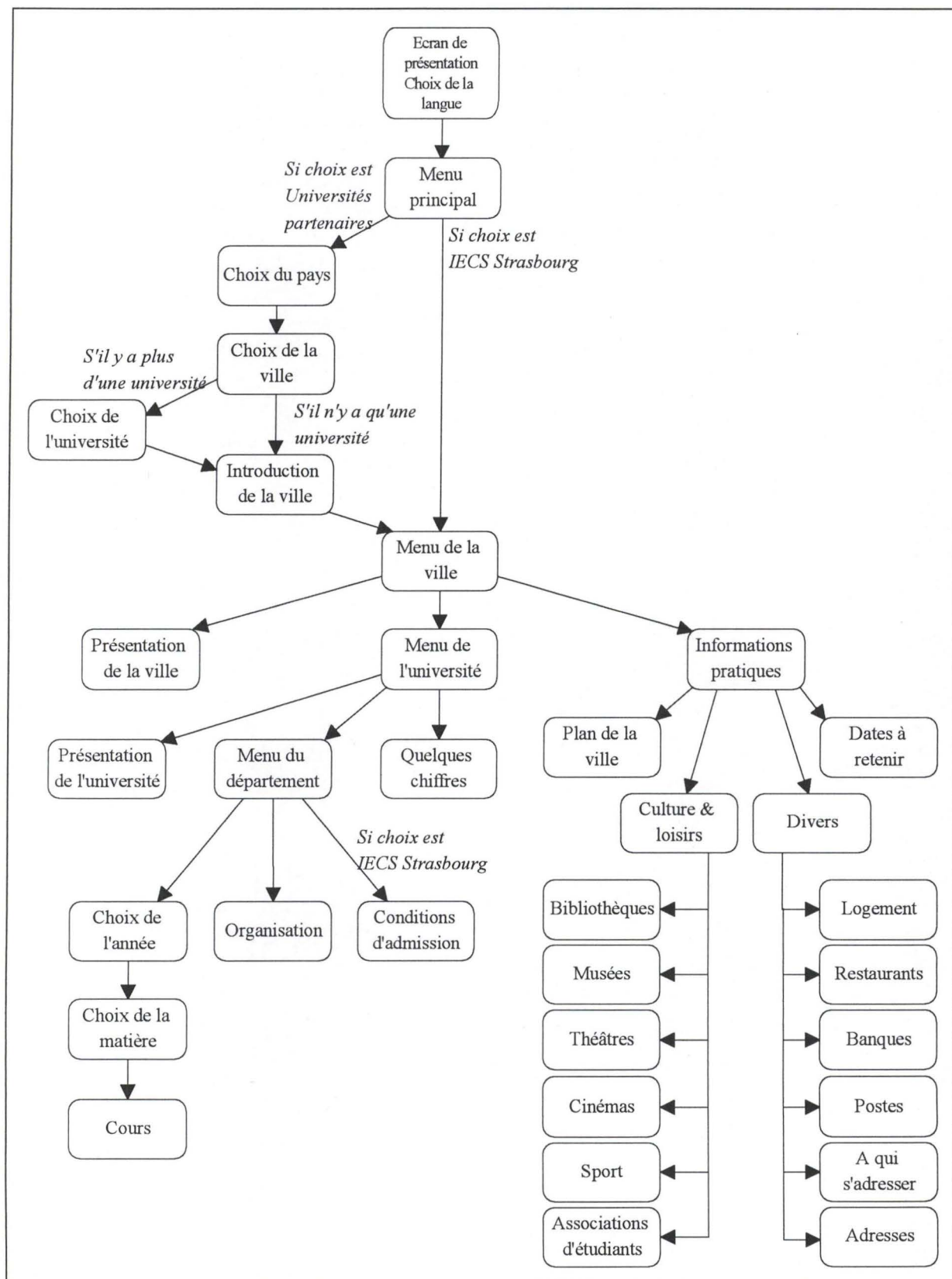


Le FEC

Place Saint-Etienne

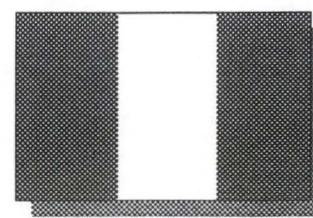
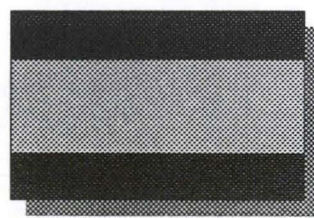
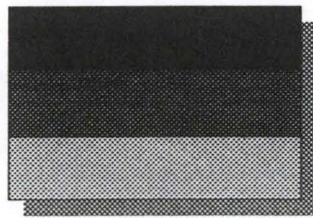
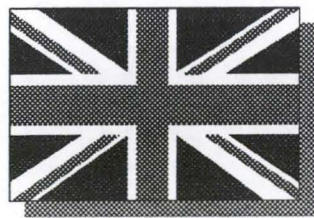
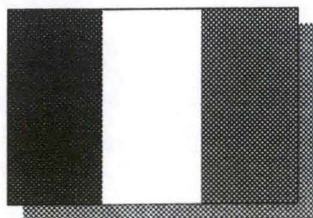
Menu graphique

Pour permettre un parcours plus aisé dans le scénario de Télémaque, nous recopions ici le graphe de transition. Celui-ci présente les liens fixes entre les différents écrans. Nous proposons, dans les pages qui suivent, la deuxième partie du scénario. Nous prenons comme hypothèse que le choix de l'utilisateur au niveau du menu principal est: Universités partenaires.





TELEMAQZIE



Menu graphique

Universités partenaires

IECS Strasbourg

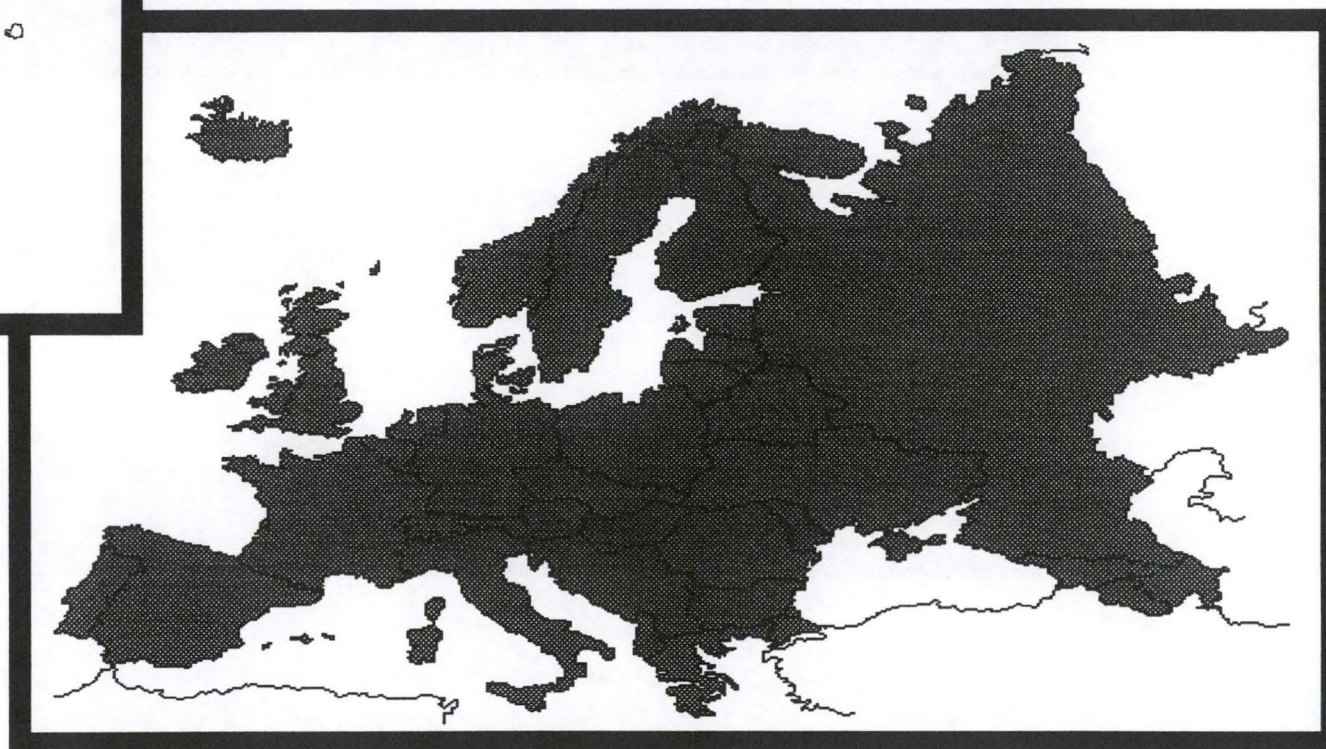
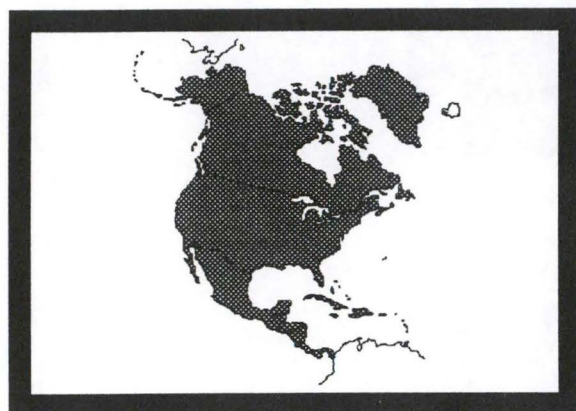
Menu principal





Universités partenaires

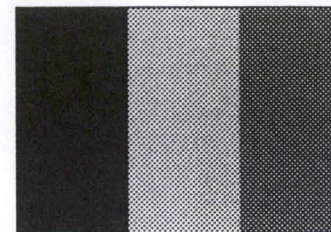
Appuyez sur le pays de votre choix



Menu graphique



Belgique

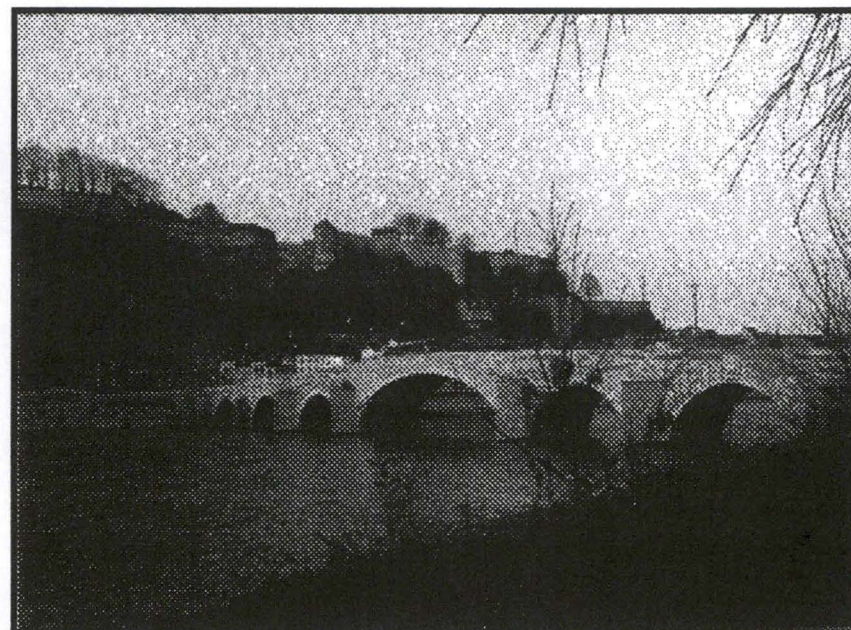
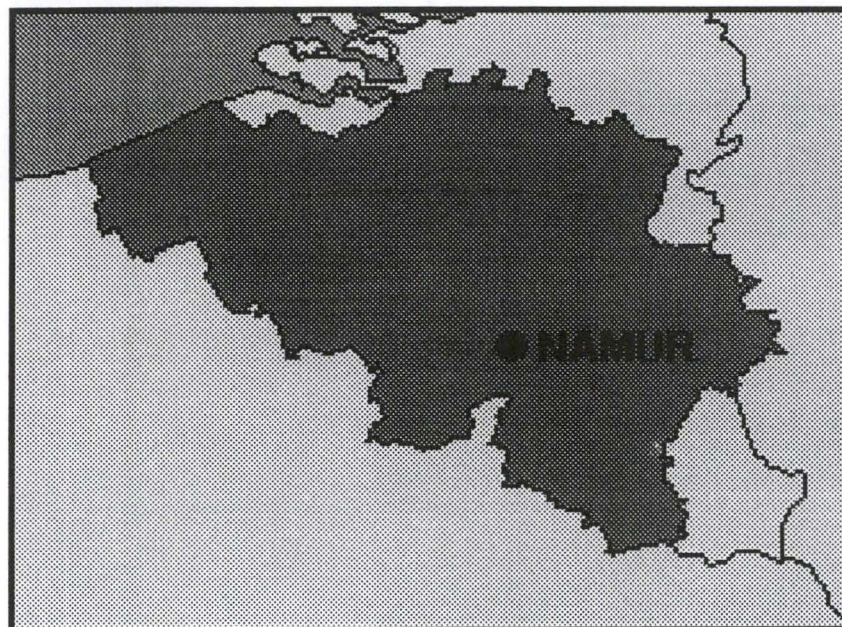


Antwerpen

Louvain-La-Neuve

Namur

Menu graphique



Namur, capitale de la Wallonie

SUITE

Menu graphique



Namur



Présentation de la ville

Informations sur l'université

Informations pratiques

Menu graphique



Namur

Présentation de la ville



ENTRE 2 ET 6 ECRANS POUR
L'HISTORIQUE DE LA VILLE



Namur

Présentation de la ville



Entre 3 et 5 écrans pour définir
les curiosités culturelles

- Citadelle de Namur
- Musée Félicien Rops
- Fête de Wallonie
- Etc.



Namur

Présentation de la ville



Au maximum 1 écran pour
présenter le rôle politique de la
ville

- Capitale de la Wallonie
- Siège du parlement wallon



Namur



Présentation de la ville

Informations sur l'université

Informations pratiques

Menu graphique





Namur

Informations sur l'université



Présentation de l'université

Quelques chiffres

Institut d'informatique

Menu graphique



Namur

Présentation de l'université



- Maximum 5 écrans pour présenter un historique de l'université (ex : ligne du temps) et sa situation actuelle (composition)
- Utilisation de texte, dessins, photos, vidéos,
...



Namur

Quelques chiffres



Statistiques sur

- ☎ l'évolution du nombre d'étudiants dans l'université
- ☎ l'évolution du nombre d'étudiants dans la faculté de gestion
- ☎ l'évolution du nombre d'étudiants dans chaque année à la faculté de gestion
- ☎ le nombre d'étudiants par professeur
- ☎ l'évolution du nombre de diplômés

Menu graphique



Namur

Institut d'informatique



Cursus

Organisation

Menu graphique



Namur
Cursus



Première année

Deuxième année

Troisième année

Menu graphique



Namur

Première année



Math

Analyse

BD

Programmation

Menu graphique



Namur Math



Théorie des graphes

Processus stochastiques

J. Fichefet

M. Noirhomme

45 h

45 h

Menu graphique



Namur
Organisation



- Organigramme de la faculté
- Nom, fonction, photo des personnes y figurant
- présentation des professeurs

Menu graphique



Namur



Présentation de la ville

Informations sur l'université

Informations pratiques

Menu graphique

Namur

Informations pratiques



Culture et loisirs

Divers

Plan de la ville

Dates à retenir

Menu graphique



Cinéma

Musées

Sport

Théâtres

Associations
d'étudiants

Bibliothèques

Menu graphique

Namur

Divers



Logements

Restaurants

Banques

Postes

A qui s'adresser

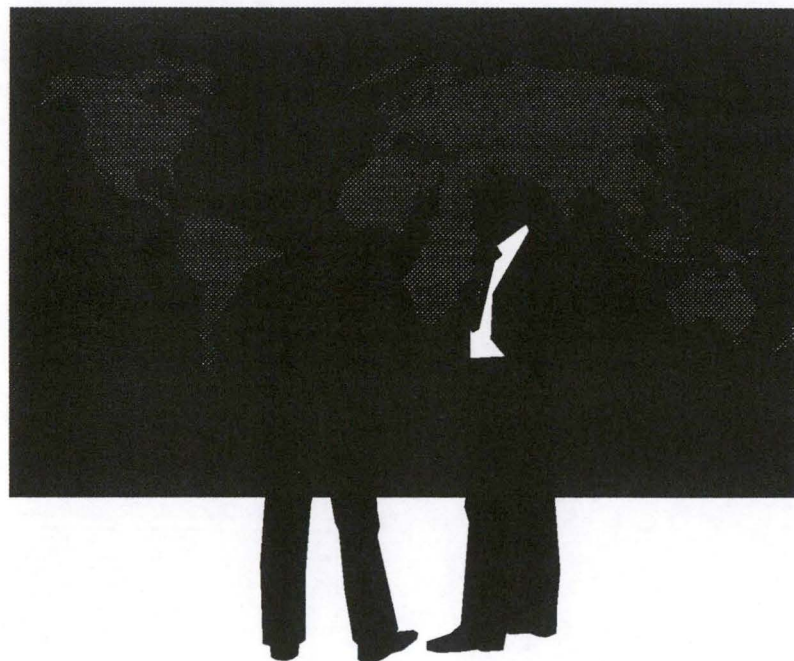
Adresses

Menu graphique



Namur

Plan de la ville



Menu graphique



Namur

Dates à retenir



Fêtes de Wallonie

3ème dimanche de septembre

Fêtes traditionnelles

Menu graphique



Namur

Restaurants



L'arsenal

Le fac-food

Rue Bruno

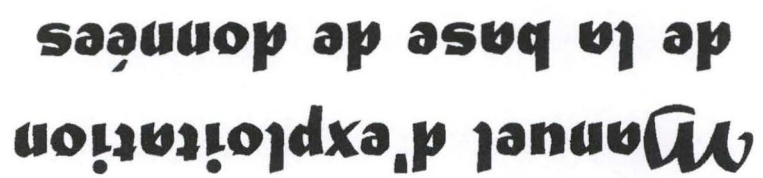
Rue Grafé

Menus à 120 et 145 FB

Menus à 120 et 145 FB

Menu graphique

ANNEXE 3. MANUEL D'EXPLOITATION DE LA BASE DE DONNÉES



312104
AW
3734

Table des matières

1. Préliminaires	2
2. Un peu de vocabulaire ...	3
2.1. Ecran ou formulaire	3
2.2. Enregistrement	3
2.3. Champ ou Attribut	4
2.4. Valeur	4
2.5. Stockage des données	4
3. Le déplacement	5
3.1. Général	5
3.1.1. Bouton doigt	5
3.1.2. Bouton stop	5
3.2. Déplacement entre formulaires	5
3.2.1. Bouton retour pays	5
3.2.2. Bouton écran suivant	5
3.2.3. Bouton écran précédent	6
3.3. Déplacement entre enregistrements	6
3.3.1. Boutons premier et dernier enregistrement	6
3.3.2. Boutons enregistrement précédent et suivant	6
3.3.3. Bouton rechercher	6
3.3.4. Bouton supprimer	6
3.3.5. Bouton nouveau	7
4. Le remplissage des écrans ou formulaires	8
4.1. Ecran pays	9
4.2. Ecran ville	9
4.3. Ecran université	10
4.4. Ecran présentation (ville)	11
4.5. Ecran infos pratiques	11
4.6. Ecran faculté	12
4.7. Ecran présentation (université)	13
4.8. Ecran graphique et graphique (suite)	13
4.9. Ecran organisation	15
4.10. Ecran cours	15
5. Comment créer un répertoire ou un sous-répertoire?	16

1. *Préliminaires*

Lorsqu'on ouvre la base de données, le système demande un nom d'utilisateur et un mot de passe. A ce moment-là, on tape *gestionnaire* comme nom d'utilisateur si l'on veut entrer de nouvelles informations sur une université.

Par contre, si on veut modifier la conception de la base de données, le nom d'utilisateur à introduire est *administrateur* et le mot de passe est *iecs*.

2. *Un peu de vocabulaire ...*

A l'aide d'un exemple, nous allons définir quelques termes qui reviendront souvent dans la suite de ce document.

The screenshot shows a software window titled "PAYS" with a menu bar (Fichier, Edition, Affichage, Enregistrements, Fenêtre, ?) and a toolbar. The main area contains a form with the following elements:

- Top left: A logo with the text "78 LE" and "AQUA".
- Top center: The title "PAYS" in large letters.
- Top right: A "STOP" button.
- Form fields:
 - "Nom du pays:" followed by a text input containing "Belgique-Luxembourg" and a dropdown arrow.
 - "Drapeau:" followed by a text input containing "dBelgiqu" and a dropdown arrow.
 - "Carte:" followed by a text input containing "cBelgiqu" and a dropdown arrow.
- Right side buttons (vertical stack): "Premier", "Dernier", "Précédent", "Suivant", "Rechercher", "Supprimer", "Nouveau".
- Bottom buttons (horizontal stack): "Accéder à l'écran VILLE", "Retour à l'écran d'INTRODUCTION".
- Status bar at the bottom: "Mode Formulaire" on the left, "NUM" on the right, and a central area showing "4" and "sur 25".

Figure 2-1. Le formulaire pays

2.1. Ecran ou formulaire

Un écran ou formulaire (celui de la Figure 2-1 s'appelle pays) comprend deux types d'informations :

- 1) des informations concernant une unité sémantique;
- 2) des moyens pour se déplacer et pour modifier les informations.

Les différentes unités sémantiques possibles sont : le pays, la ville, l'université, la présentation de la ville, les informations pratiques, la faculté, la présentation de l'université, le graphique, le graphique (suite), le cours et l'organisation.

2.2. Enregistrement

Un enregistrement est une des valeurs d'une unité sémantique.

Exemple : l'ensemble $\{Belgique-Luxembourg, dbelgiqu, cbelgiqu\}$ est un enregistrement de l'unité sémantique **pays**.

2.3. Champ ou Attribut

Un champ ou attribut est une caractéristique d'une unité.

Exemple : **drapeau** est un champ (ou attribut) de l'unité sémantique **pays**.

2.4. Valeur

Une valeur est le contenu attribué à un champ.

Exemple : *Belgique* est une valeur affectée au champ **Nom du pays**.

2.5. Stockage des données

Nous allons voir dans ce point les différentes notions qui ont trait à l'enregistrement des données.

La **racine** du disque est la base sur laquelle vont venir se greffer les répertoires. Elle a comme nom généralement une lettre comprise entre C et Z.

Ce que nous appelons **répertoire** est en fait un dossier « électronique » mis directement sur la racine. C'est à dire qu'il s'agit d'un endroit dans lequel on va pouvoir classer des informations (dans notre cas, ces informations seront des photos). Un sous-répertoire étant simplement un répertoire posé dans un autre répertoire.

L'**arborescence** du disque est l'ensemble de tous les répertoires contenus sur un disque, avec leur hiérarchie.

Par exemple, C:\

```
Telemaqu
  Albanie
  Allemagn
  Belgique
    Antwerpe
    Namur
      Univ
      Ville
Windows
  System
  Msapps
  Twain
```

est un exemple (incomplet...) de l'arborescence du disque C.

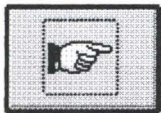
Dans cet exemple, la racine est C. Un répertoire est par exemple *Telemaqu*. Un sous-répertoire de *Telemaqu* est par exemple *Albanie*, un autre étant *Belgique*. On appelle aussi *Namur*, le répertoire **Père** des sous-répertoires *Univ* et *Ville*. Ces deux-ci étant appelés les répertoires **Fils**.

3. *Le déplacement*

Le déplacement dans l'ensemble des types d'écrans ou formulaires se fait au moyen de boutons expliqués dans les points qui suivent.

3.1. Général

3.1.1. Bouton doigt



Permet de passer de l'écran d'introduction à l'écran pays

3.1.2. Bouton stop



*Permet de quitter la base de données **TELEMAQUIE***

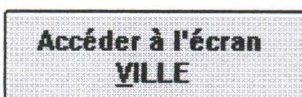
3.2. Déplacement entre formulaires

3.2.1. Bouton retour pays



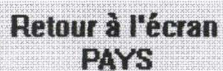
Permet de retourner au formulaire pays

3.2.2. Bouton écran suivant



Permet d'accéder à un des formulaires qui dépendent du formulaire courant

3.2.3. Bouton écran précédent



Retour à l'écran
PAYS

Permet de retourner au formulaire précédent

3.3. Déplacement entre enregistrements

3.3.1. Boutons premier et dernier enregistrement



Premier

Dernier

Permet de se placer sur le premier ou le dernier enregistrement de l'unité courante

3.3.2. Boutons enregistrement précédent et suivant



Précédent

Suivant

Permet de se placer sur l'enregistrement qui précède ou qui suit l'enregistrement courant

3.3.3. Bouton rechercher



Rechercher

Permet de trouver, dans l'ensemble des enregistrements d'une unité, une suite de caractères introduits par le biais du clavier

3.3.4. Bouton supprimer

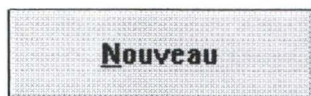


Supprimer

Permet de supprimer l'enregistrement courant

Attention : si des enregistrements dépendent de l'enregistrement supprimé (en d'autres termes, s'ils existe au moins un bouton écran suivant), ceux-ci vont également être détruits.

3.3.5. Bouton nouveau



Permet de créer un enregistrement vide dans lequel on va pouvoir entrer de nouvelles valeurs

4. Le remplissage des écrans ou formulaires

Ce point est consacré au remplissage proprement dit. Nous allons passer en revue les différents écrans et mentionner, pour chacun, les règles à respecter.

Comme on le constatera, les informations introduites dans la base de données sont de deux types: du texte et des références à des photos. Si le premier type d'information est entièrement géré par le gestionnaire de la base de données, ce n'est pas le cas pour le second. Dans les différents points qui suivent, nous indiquerons dans quel répertoire du disque dur il faudra mettre les informations correspondantes.

4.1. Ecran pays

Lorsqu'on introduit une nouvelle université partenaire, deux cas sont possibles:

- le pays dans lequel elle se trouve existe déjà
=> on peut passer à l'écran **ville**;
- le pays dans lequel elle se trouve n'existe pas encore
=> suivre la procédure qui suit.

Pour donner une valeur au champ **Nom du pays**, on peut soit l'introduire au clavier soit choisir un des noms proposés dans la liste déroulante. Dans les deux cas, on ne peut pas introduire un nom ne figurant pas dans la liste.

Si le pays qu'on veut ajouter n'est pas proposé, il faut donner la valeur *Reste du monde* (valeur « fourre-tout »).

Lorsqu'on aura donné une valeur à ce champ, on verra que les champs **Drapeau** et **Carte** prendront automatiquement les valeurs adéquates.

On peut voir un écran Pays à la Figure 4-1.

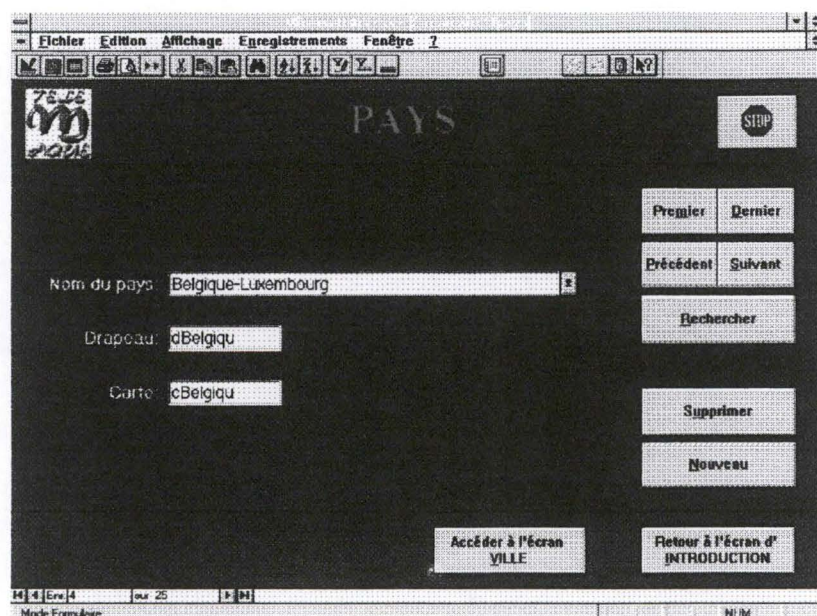


Figure 4-1. Ecran Pays

4.2. Ecran ville

Il faut remplir les différents champs en veillant à ce que deux villes différentes n'aient pas le même **Numéro**.

Si on crée une nouvelle ville, il faut également créer un sous-répertoire du pays courant et lui donner comme nom les huit premiers caractères du nom de la ville (si la ville comporte moins de huit caractères, il n'y a pas de problème). Dans ce sous-répertoire, vous devez en créer deux que l'on nommera *Univ* et *Ville*. Dans le répertoire créé avec les huit premiers caractères du nom de la ville, il faudra y mettre les informations reprises dans le formulaire Ecran ville (c'est-à-dire les fichiers repris sous les champs **logo**, **photo gauche**, **photo droite**, **plan**).

Pour créer un répertoire ou un sous-répertoire, voir le point 5 de ce document.
Un exemple d'écran ville est présenté à la Figure 4-2.

78-LE
VILLE
Belgique-Luxembourg

Nom de la ville: Namur

Numéro: 3

Logo: namlogo

Plan:

Photo(gauche): ptitrain

Photo(droite): citadel

Phrase: Namur, capitale de la Wallonie

Premier Dernier

Précédent Suivant

Rechercher

Supprimer

Nouveau

Accéder à l'écran UNIVERSITE

Accéder à l'écran PRESENTATION

Accéder à l'écran INFOS PRATIQUES

Retour à l'écran PAYS

4 Enc:3 sur 3

Mode Formulaire

NUM

Figure 4-2. Ecran Ville

4.3. Ecran université

Il n'y a rien de particulier à signaler pour cet écran. On peut en consulter un à la Figure 4-3.

- Le fichier repris sous le champ **Logo** doit être placé dans le répertoire *Univ* de l'université concernée.

78-LE
UNIVERSITE
Namur

Nom de l'université: FUNDP

Logo: logfndp

Premier Dernier

Précédent Suivant

Rechercher

Supprimer

Nouveau

Accéder à l'écran FACULTE

Accéder à l'écran PRESENTATION

Accéder à l'écran GRAPHIQUE

Retour à l'écran VILLE

4 Enc:1 sur 1

Mode Formulaire

NUM

Figure 4-3. Ecran Université

4.4. Ecran présentation (ville)

Il faut introduire une **Phrase** (en 5 langues) et la **Photo** qui lui correspond.

Le **Numéro** est, lui, affecté automatiquement. On a le droit de modifier le **Numéro** d'un nouvel enregistrement mais pas celui d'un enregistrement existant. On peut voir à la Figure 4-4 un écran du type Présentation-ville.

Exemple

Si on a déjà 12 enregistrements, on peut en créer un nouveau et lui donner le numéro 6. Tous les numéros seront décalés de façon à insérer le nouvel enregistrement au bon endroit. Par contre, on ne peut pas changer l'enregistrement numéro 5 et lui donner la valeur 3.

- Le fichier repris sous le champ **Photo** doit être placé dans le répertoire *Ville* de la ville concernée.

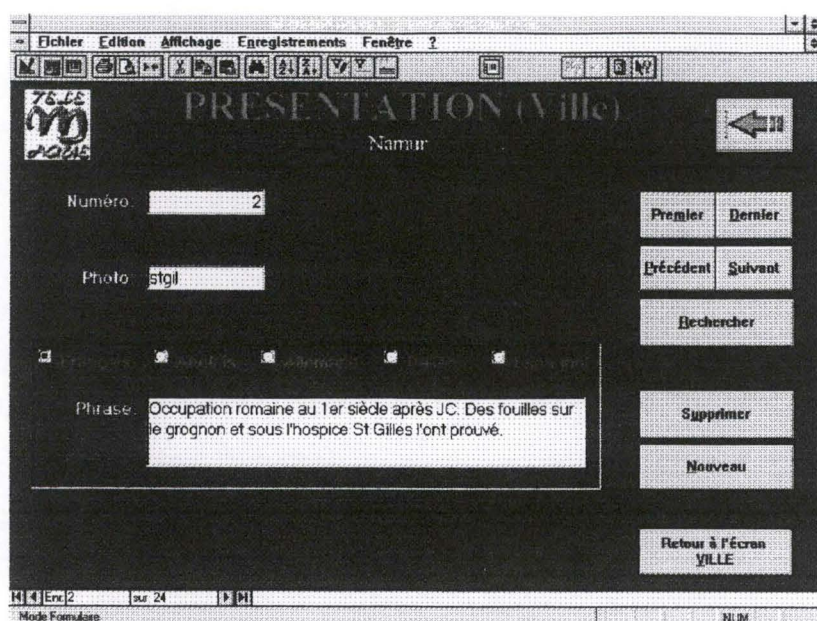


Figure 4-4. Ecran Présentation-ville

4.5. Ecran infos pratiques

Comme pour l'écran pays, il faut choisir un **Type** d'informations dans la liste proposée.

Lorsque ce **Type** est *Dates à retenir*, on doit entrer la **Date de début** et la **Date de fin**. Ces deux dates respectent le format JJ/MM. A la Figure 4-5, on retrouve un écran Infos pratiques.

Figure 4-5. Ecran Infos pratiques

4.6. Ecran faculté

Il faut entrer le **Nom de la faculté** et le **Nombre d'années** que dure son cycle d'études. On trouve à la Figure 4-6 un exemple de ce type d'écran.

Figure 4-6. Ecran faculté

4.7. Ecran présentation (université)

Mêmes remarques que pour l'écran **présentation-ville**. On peut voir cet écran à la Figure 4-7.

- Le fichier repris sous le champ **Photo** doit être placé dans le répertoire *Univ* de l'université concernée.

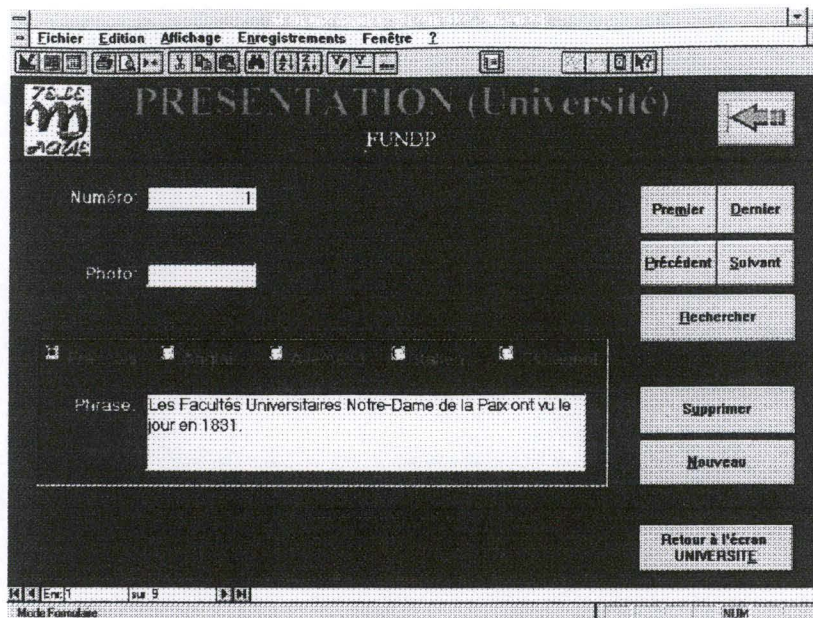


Figure 4-7. Ecran Présentation-université

4.8. Ecran graphique et graphique (suite)

Il faut introduire d'abord le **Titre** du graphique, le **Type** du graphique (à choisir dans la liste qui se compose des éléments suivants: aire, histogramme et ligne) et la suite d'**Abscisses** de ce graphique.

Ensuite, il faut passer à l'écran **graphique (suite)** où on va remplir les champs **Légende** et **Ordonnées**.

Exemple

Nous allons créer un graphique reprenant l'évolution du nombre d'étudiants. Voici les valeurs à introduire.

Ecran graphique:

- Nom du graphique: Evolution du nombre d'étudiants
- Type du graphique: Histogramme
- Abscisses: 1990,1991,1992,1993

Ecran graphique (suite):

enregistrement 1

- Légende: Faculté d'économie
- Ordonnées: 200,225,260,230

enregistrement 2

- Légende: Faculté de droit
- Ordonnées: 312,295,250,300

enregistrement 3

- Légende: Faculté de biologie
- Ordonnées: 50,70,72,65

Les deux écrans graphique et graphique (suite) sont visibles à la Figure 4-8 et à la Figure 4-9.

Figure 4-8. Ecran Graphique

Figure 4-9. Ecran Graphique (suite)

4.9. Ecran organisation

Il n'y a rien à signaler à propos de cet écran. On peut en consulter un à la Figure 4-10.

The screenshot shows a window titled 'Fichier Edition Affichage Enregistrements Fenêtre 2'. The main area has a dark background with the title 'ORGANISATION' and 'Institut d'informatique'. Below the title, there are several tabs: 'Fonction', 'Année', 'Groupe', 'Titre', and 'Cours'. The 'Fonction' field is set to 'Directeur' and the 'Nom' field is set to 'R. Lesuisse'. On the right side, there are buttons for 'Premier', 'Dernier', 'Précédent', 'Suivant', 'Rechercher', 'Supprimer', 'Nouveau', and 'Retour à l'écran FACULTE'. The status bar at the bottom indicates 'Mode Formulaire' and 'NUM'.

Figure 4-10. Ecran Organisation

4.10. Ecran cours

La valeur que l'on affecte au champ **Année** ne peut pas être supérieure à la valeur de **Nombre d'années** de l'écran **faculté**. Le **Groupe** d'un cours est la branche dans laquelle il se situe (par exemple, gestion, comptabilité, informatique, etc.). On peut voir un écran cours à la Figure 4-11.

The screenshot shows a window titled 'Fichier Edition Affichage Enregistrements Fenêtre 2'. The main area has a dark background with the title 'COURS' and 'Institut d'informatique'. Below the title, there are several tabs: 'Année', 'Intitulé', 'Groupe', 'Professeur', and 'Nombre d'heures'. The 'Année' field is set to '1', the 'Intitulé' field is set to 'Théorie des graphes', the 'Groupe' field is set to 'Mathématiques', the 'Professeur' field is set to 'J. Fichet', and the 'Nombre d'heures' field is set to '45'. On the right side, there are buttons for 'Premier', 'Dernier', 'Précédent', 'Suivant', 'Rechercher', 'Supprimer', 'Nouveau', and 'Retour à l'écran FACULTE'. The status bar at the bottom indicates 'Mode Formulaire' and 'NUM'.

Figure 4-11. Ecran Cours

5. Comment créer un répertoire ou un sous-répertoire?

Pour créer un répertoire (ou un sous-répertoire, la procédure étant la même, nous ne ferons dorénavant plus la distinction), il faut utiliser le *Gestionnaire de fichiers*, situé dans le *Groupe principal* de Windows. Lorsque le *gestionnaire de fichiers* est lancé, il faut ouvrir le disque dur C (c'est à dire celui sur lequel se trouve Télémaque, dans notre cas). Pour créer un répertoire, il suffit de se positionner à l'endroit voulu sur l'arborescence du disque. Ensuite, il faut aller dans le menu *Fichier* et choisir l'item *Créer un répertoire*. A ce moment, on verra apparaître une boîte de dialogue demandant de rentrer le nom du nouveau répertoire. Il faut alors le rentrer et valider cette opération en appuyant sur le bouton *OK*.

Pour clarifier cette procédure qui peut paraître complexe, créons le répertoire *Bonn*, comme répertoire fils de *Allemagn* dans notre exemple ci-dessus. Pour ce faire, on doit cliquer sur le répertoire *Allemagn* (répertoire père) et ensuite aller dans le menu *Fichier*, pour y choisir l'item *Créer un répertoire*. Là, il faut introduire le nom de notre nouveau répertoire, *Bonn*.

Après cette opération, voici ce que l'on verra dans la fenêtre reprenant l'arborescence :

```
C:\
  Telemaqu
    Albanie
    Allemagn
      Bonn
    Belgique
      Antwerpe
      Namur
        Univ
        Ville
  Windows
    System
    Msapps
    Twain
```